Db2 Web Query BI Portal

Release 2.3

Contents

Chapter 1

Introducing Db2 Web Query for i

This documentation provides an overview of Db2 Web Query for i. It also describes the components and tools you use to run, view, create, and edit reports. Use this documentation to learn about the structure and the capabilities of each component and tool.

In this chapter:

Db2 Web Query for i Overview

Db2 Web Query for i is a complete, web-ready data access and reporting system, which takes advantage of the low-cost, low-maintenance, and wide distribution capabilities of the World Wide Web and internal corporate websites. This system also provides a streamlined reporting environment that virtually eliminates the complexities of today's corporate data.

Reporting Concepts

The reporting environment includes top-level folders which are the highest level of organization. Top-level folders provide data on a particular topic (such as sales, inventory, or finance). The data is stored in different forms in the following domain tree components: Reports and Other Files. The Other Files folder, available to administrators and developers, can contain images (.gif, .jpg), HTML files, and other files that you can use to customize and enhance reports.

Reporting Features

InfoAssist+ is a Rich Internet Application (RIA) that uses AJAX (Asynchronous JavaScript and XML) technology. This powerful ad hoc reporting tool enables rapid and efficient design and deployment of reports and charts. It uses an interactive and fully customizable WYSIWYG (What You See Is What You Get) development environment. Users receive instant feedback throughout the development process to ensure that reports and charts are properly built.

Note: IBM supports only reports specifically developed in Web Query.

Reporting Interface

The reporting interface is ideal for users who quickly need to run reports. Check your installation manual for information about browser compatibility. This interface offers you the ability to access InfoAssist+.

The reporting interface is an HTML-based front-end that allows you to:

- Run reports.
- Create reports and graphs using InfoAssist+.
- Save reports and graphs.
- Share reports with other users.
- Edit reports.
- Manage Metadata.

Development Environment

The Db2 Web Query Developer Workbench product is an optional add-on Windows-based development environment for creating applications. Developer Workbench provides intuitive GUI tools that free developers from the hassles of coding, thereby allowing them to concentrate on interface design, business logic, and data manipulation. Using Developer Workbench, developers can build powerful web page interfaces that allow users to run customized reports.

Security Concepts

The following outlines role-based security concepts.

Web Query Groups

Web Query Groups are pre-defined to represent a specific set of functions or role. Global Groups define a role at the Web Query product level and apply across folders, whereas folder Groups define a role at the top-level folder level.

There are two global Groups defined as follows.

Group Name	Role Description
WebQueryAdministrator	Can perform all functions in Web Query and can access all folders.
DevWorkBench	Can connect to Web Query using Developer Workbench.

The six folder Groups are defined as follows.

Group Name	Role Description
Folder-run	Can run procedures in the respective folder.
Folder-analyst	Folder-run role plus the ability to develop and run procedures in private folders.
Folder-dev	Folder-analyst role plus the ability to develop, run, and publish procedures in published folders within the respective folder.
Folder-dba	Can manage metadata in the respective application folder.
Folder-sched	Can manage schedules and distribution lists in the respective folder.
Folder-admin	Can manage users in the respective folder. **Note:** A Folder-admin cannot acquire or release a developer or group profile license.

Each top-level folder that is created will automatically have the six folder based groups created in the Web Query Repository.

For example, if you add a top-level folder named *Sales*, the following six groups are automatically created:

- **Sales-run**. Can run reports in the Sales folder.
- **Sales-analyst**. Can develop and run reports in private folders within the Sales folder.
- **Sales-dev.** Can develop, run, and publish reports in a published folder within the Sales folder.
- **Sales-dba.** Can manage metadata in the Sales folder application directory.
- **Sales-sched.** Can manage schedules and distribution lists in the Sales folder.
- **Sales-admin.** Can add a user to or remove a user from the Sales folder group.

Users are added to one or more groups to provide the functionality they require to perform their job. This is done using the Security Center. For more information on how to add users to Web Query groups, see *Security Center* on page 15.

Chapter 2

Using the Reporting Interface

When you enter the reporting interface, you are automatically connected to Db2 Web Query for i.

From the reporting interface, you can:

- Navigate the Repository tree.
- Right-click any item to perform different functions on that item.
- Access InfoAssist+ to develop reports, charts, documents, or visualizations.

Note: The browser Back and Forward buttons cannot be used to navigate between pages in the interface, or from an interface page to a page viewed before connecting to the interface.

In this chapter:

Recommended Browser Setup

The following browser setup is recommended for use with the interface:

- ❑ Do not cache page content.
- ❑ Disable pop-ups for reporting environment.

For information on browser setting options, see the browser Help system.

Signing in to Db2 Web Query

The port number in the URL is *always* 12331. This is the pre-configured port for Db2 Web Query 2.2.x.

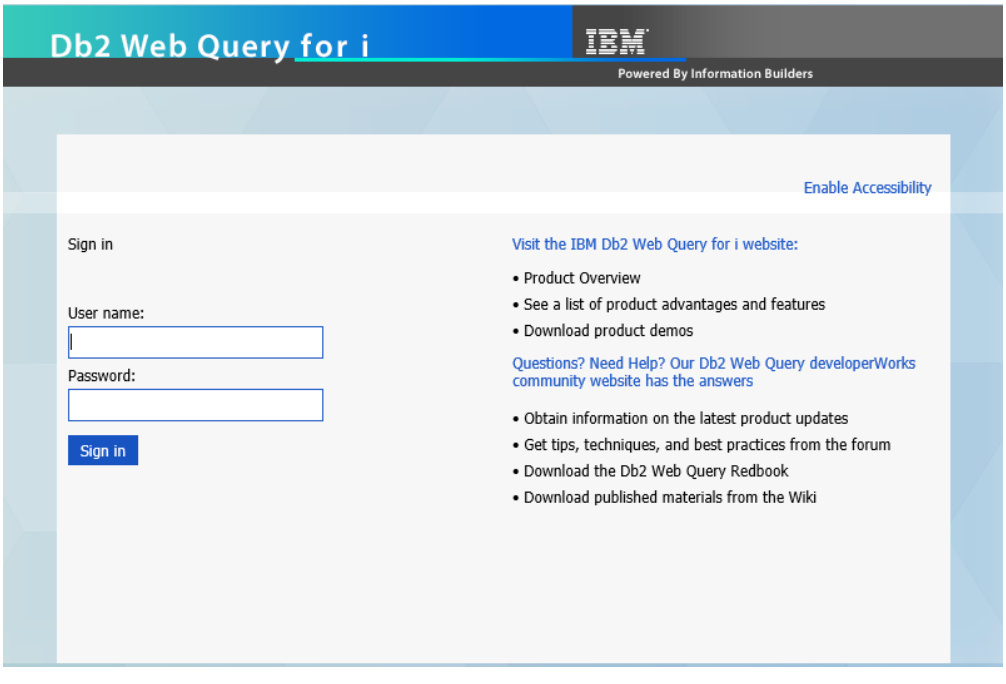
Procedure: How to Sign in to Db2 Web Query

In the following task, the system name is MYSYSTEM.ABC.ACME.COM:

1. Type the following internal URL:

   `http://mysystem.abc.acme.com:12331/webquery`

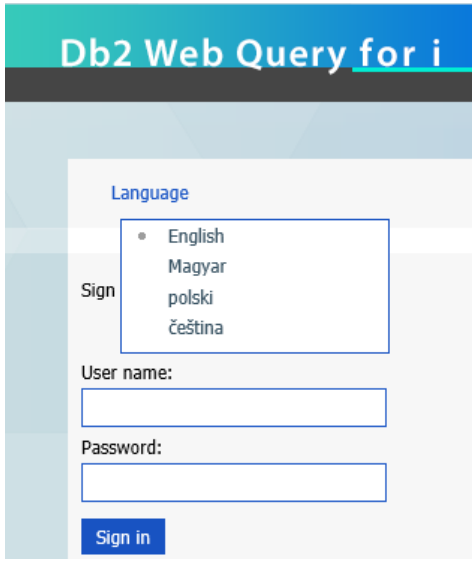
   The Db2 Web Query for i Sign in page opens, as shown in the following image.



2. Sign in to Db2 Web Query with the same user name and password that you use to access the IBM i machine.

   Tip: Add your Db2 Web Query URL to your browser Favorites.

   If your Db2 Web Query instance was configured with Dynamic Language selection, your sign in page provides a Language drop-down menu, as shown in the following image.



Web Query Folders

Almost all content is stored in the Repository, as shown in the following image. The only exception is Web Query metadata. Synonyms are still stored in the IFS in application directories.

Web Query content consists of procedures, HTML files, Stylesheets, images, Report Broker schedules, and distribution lists. This content is stored in one or more folders.

Top-Level Folders

Top-level folders reside at the top of the Repository and are used to segregate applications. For example, you may want to have a top-level folder for Sales and another folder for Human Resources (HR) where users can be assigned to one or more folders. A user can have a different set of roles for each folder. For example, a user can be assigned to only run reports in HR but takes on a developer/dba role in Sales.

Each top-level folder is created with its own set of Web Query Groups that define the authorization rules for the folder.

The Common top-level folder exists for all Web Query installations. The purpose of this folder is to contain content that can be run by all users. If you prefer not to display the Common top-level folder, there is an option to hide it.

Procedure: How to Hide the Common Top-Level Folder

1. Sign in to Web Query using a Web Query Administrator user ID.
2. Right-click the *Common* top-level folder and click *Hide*.

Application Directories and Metadata Management

Application directories are where synonyms are created. They are IFS directories which map to the following path:

```
/qibm/UserData/qwebqry/apps
```

In Web Query release 1.1.x, the baseapp application directory is the default location for newly created synonyms. Synonyms in the baseapp directory are accessible from all folders. If Developer Workbench was licensed, it could be used to create new application directories, which could be linked to a folder application directory path.

In releases 2.1.x and 2.2.x, the baseapp application directory fulfills the same role as in release 1.1.x. However, a new application directory is created for every top-level folder and is automatically linked to the folder as the 1st directory of the application directory path.

The application directory path is searched whenever a synonym is required to:

1. Develop a new procedure.
2. Edit an existing procedure.
3. Run an existing procedure.

The automatic creation of this application directory and making it first in the application directory path allows you to segregate synonyms without using Developer Workbench. Synonyms that only pertain to one top-level folder or application should be created in that folder application directory. Synonyms that need to be shared across all applications should be copied or created in the baseapp directory.

Note: If a synonym with the same name exists in both application directories of the application directory path, the first one found in the path will be used.

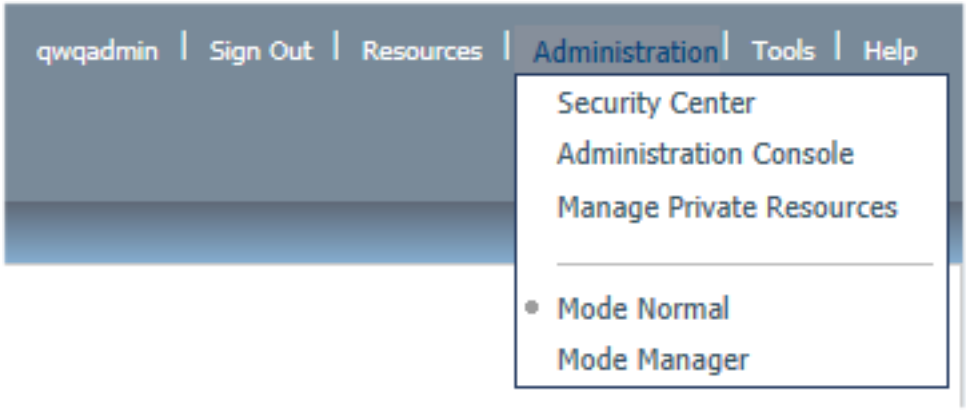
Security Center

User Management is handled through the Web Query Security Center. This is a graphical user interface that simplifies Web Query user administration because it is tightly integrated with License Manager and will dynamically update license information.

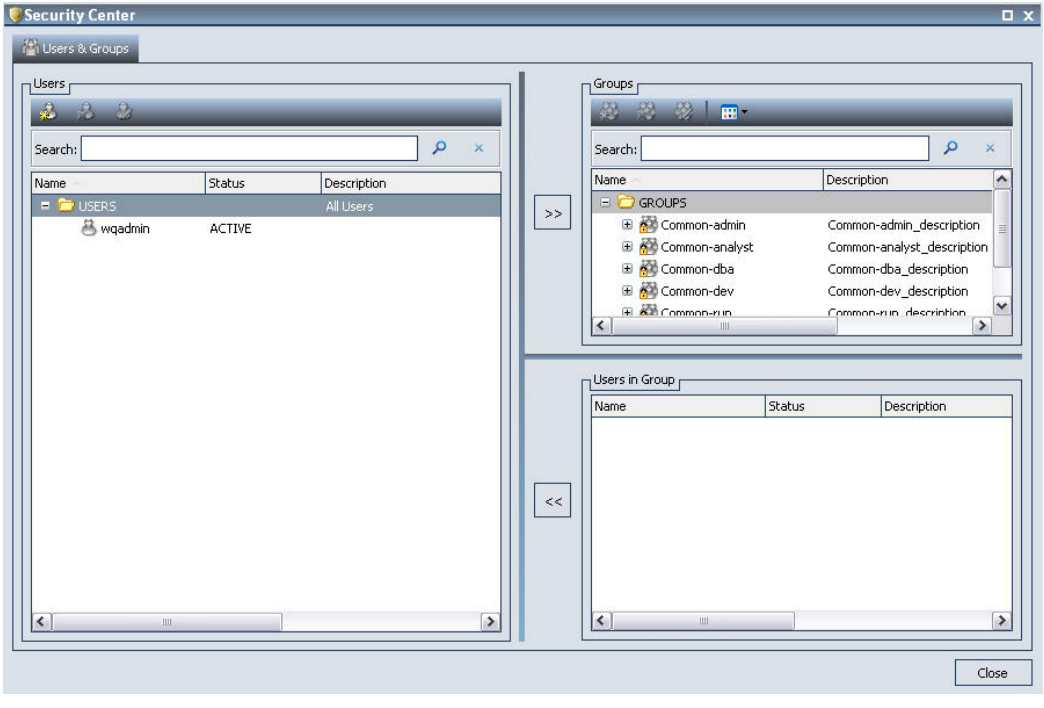
Procedure: How to Launch the Security Center

1. Sign in to Web Query using a Web Query Administrator user ID.

2. On the menu bar, right-click *Administration*, and select *Security Center*, as shown in the following image.



The Security Center opens, as shown in the following image.

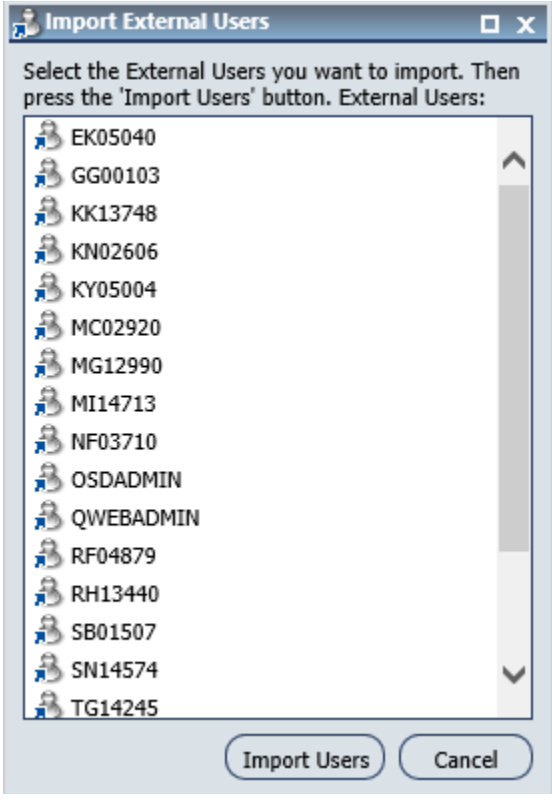


Procedure: How to Create a User

Once you add a new folder, you can create users and add them to a group in that folder.

1. Launch the Security Center.

2. Click the *New User*  icon. The Import External Users box opens, as shown in the following image.



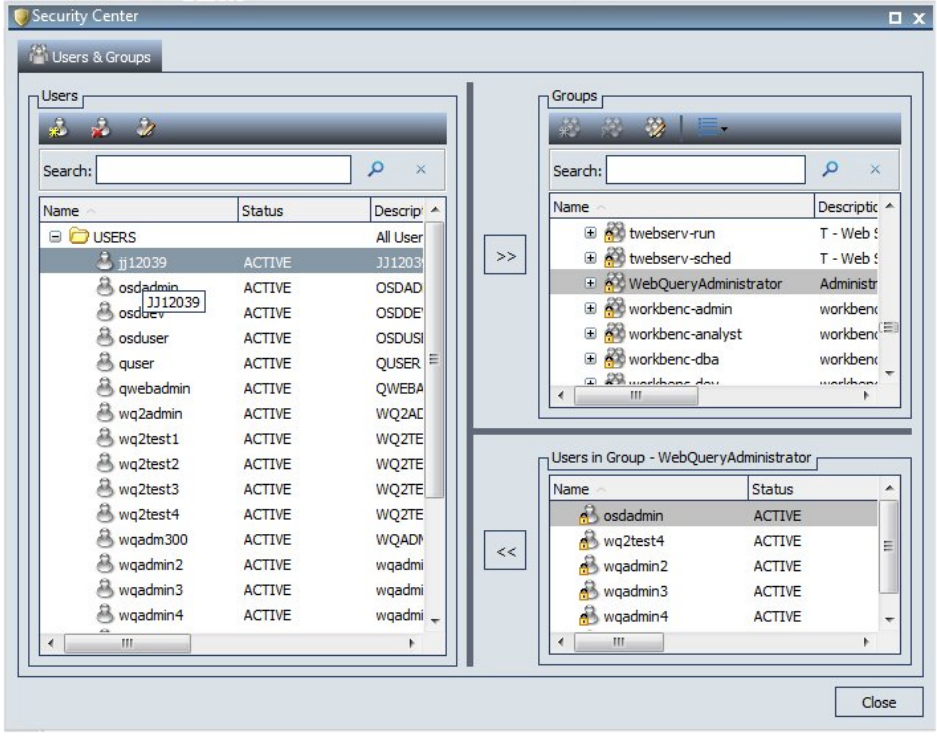
3. Select one or more users from the list and click the *Import Users* button to add the user IDs as a named user (developer) to Web Query 5733WQX.

Procedure: How to Designate a Web Query Administrator

The first task in user management is to designate a Web Query Administrator. This procedure shows how to accomplish this using the QWQADMIN user ID and the Security Center interface. Alternatively, an authorized user can use the QWEBQRY/REGWQUSR command. For more information on the QWEBQRY/REGWQUSR command, see the *Summary of New Features* manual.

Web Query Administrator user IDs are able to perform any function in Web Query, except designating and removing a Web Query Administrator.

1. Launch the Security Center.
2. In the Users panel, select the user or users to add to the group.
3. In the Groups panel, select the *WebQueryAdministrator Group*, as shown in the following image.



4. Click the *Add selected users to group* button to add the user to the group.

   Tip: You can also add users by dragging the selected users into a group.

5. Click *Close* to exit the Security Center.

   The owner of this newly added Web Query Administrator user ID can now sign in and manage the Web Query environment:

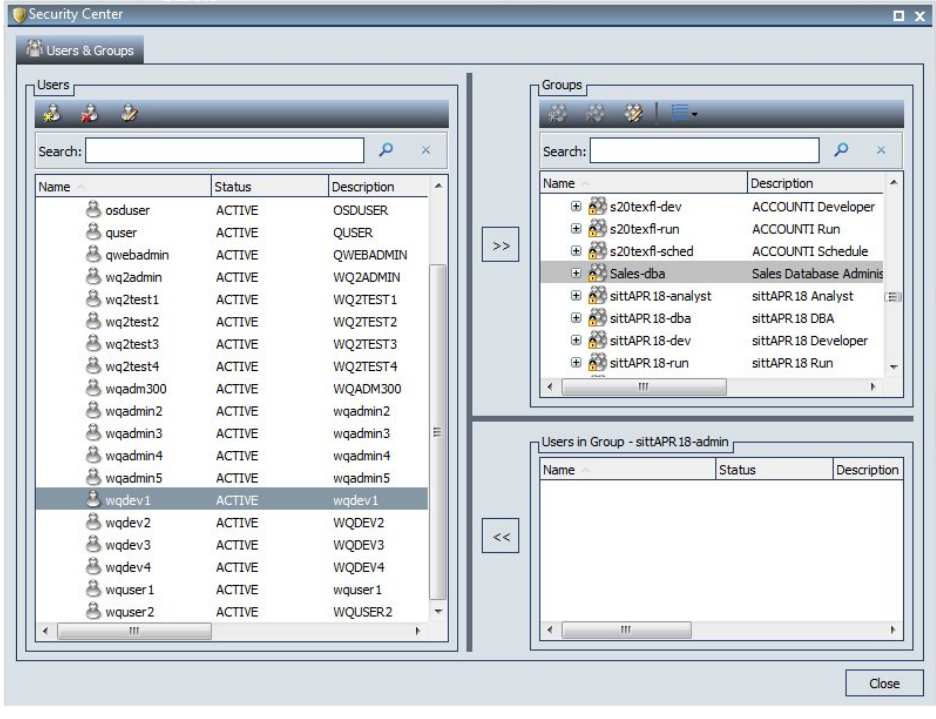
   - ❏ Create top-level folders.
   - ❏ Launch Security Center to add users and assign them to folder groups.

Procedure: How to Add a User to a Group

This example assumes a Sales top-level folder exists in Web Query. The Web Query Administrator user ID or a user ID in the Sales-Admin group can add users to one or more of the six folder groups for the Sales top-level folder.

1. Launch the Security Center.
2. In the Users panel, select the user or users to add to the group.
3. In the Groups panel, select one or more groups that the user or users will be added to.

   In the following image, the user wqdev1 is being added to the Sales-dba group. This will make the user a DBA within the Sales application folder.

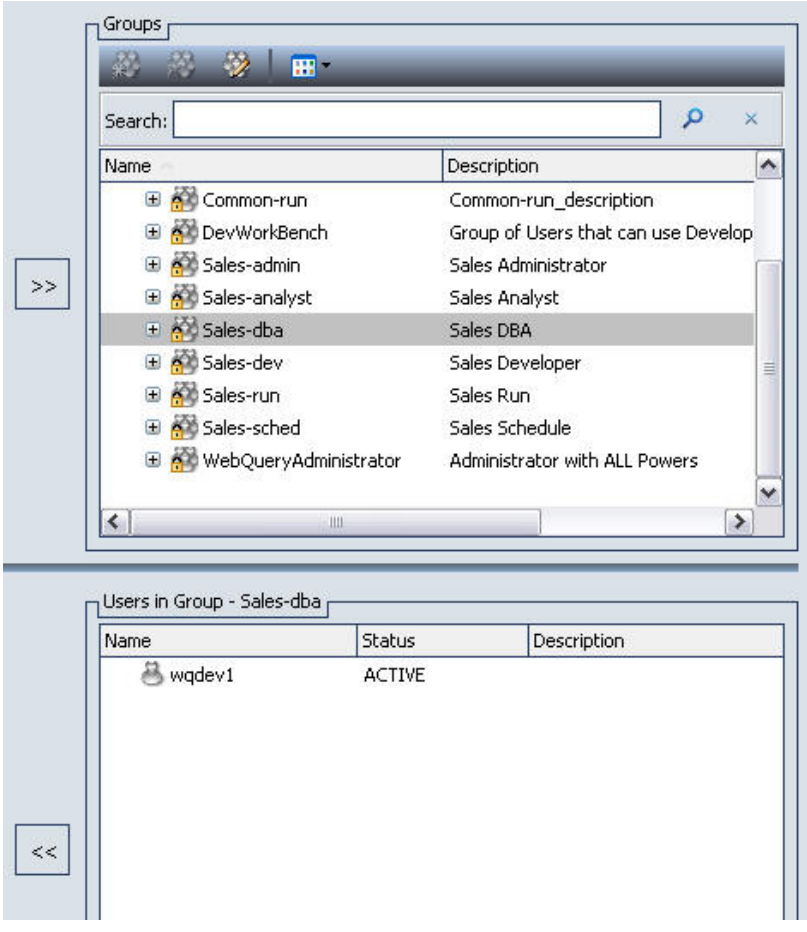


4. Click the *Add selected users to group* button to add the user to the group.

   Tip: You can also add users by dragging the selected users into a group.

5. Click *Close* to exit the Security Center.

The wqdev1 user is now in the Sales-dba group, as shown in the following image.



Adding an IBM i Group Profile to a Web Query Folder-run Group

IBM i group profiles are assigned to authorize a user for the Web Query Runtime Enablement feature. This section describes how to add an IBM i group profile to a folder-run group.

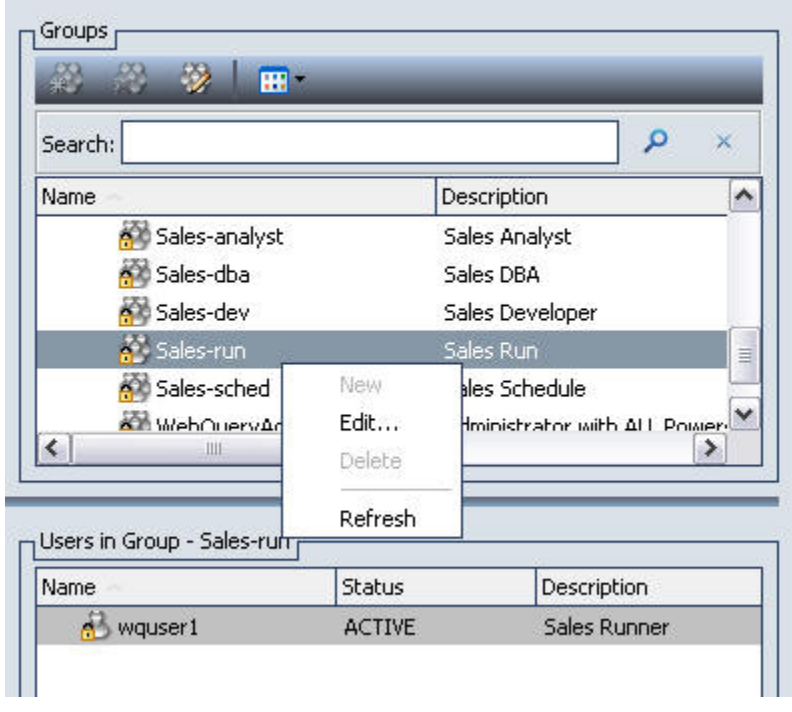
Note: The migration procedure performs this procedure for any group profiles that were assigned to users in Release 1.1.x.

Procedure: How to Assign an IBM i Group Profile to a Folder-run Group

This procedure assumes a Sales top-level folder and the objective is to assign the MUSALES group profile to the Sales-run group to allow users with this group profile to run reports in the Sales top-level folder.

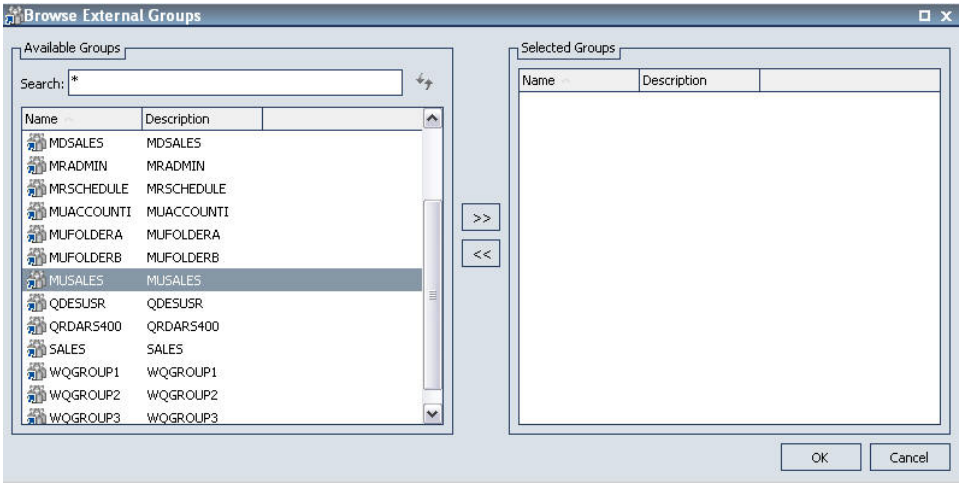
Note: Any existing group profile can be assigned to a folder-run group role. The group profile does not have to be an MUxxxxx group profile.

1. Sign in as a Web Query Administrator or folder administrator.
2. Launch the Security Center.
3. In the Groups panel, select the folder-run group that should be associated with the IBM i group profile.
4. Click the *Edit Group* button or right-click the folder-run group and select *Edit*, as shown in the following image.



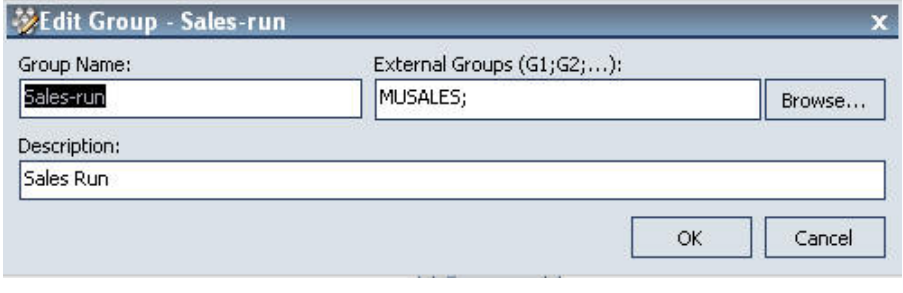
The Edit Group dialog box appears.

5. Click the *Browse* button to retrieve a list of IBM i group profiles, as shown in the following image.



6. Click the *Add all selected Groups* button to add the selected group profile to the Web Query folder-run group. In this example, the IBM i group profile MUSALES is being added to the Web Query Sales-run group.

7. Notice a Group is now associated with the Web Query folder-run group, as shown in the following image.



8. Click *OK*.

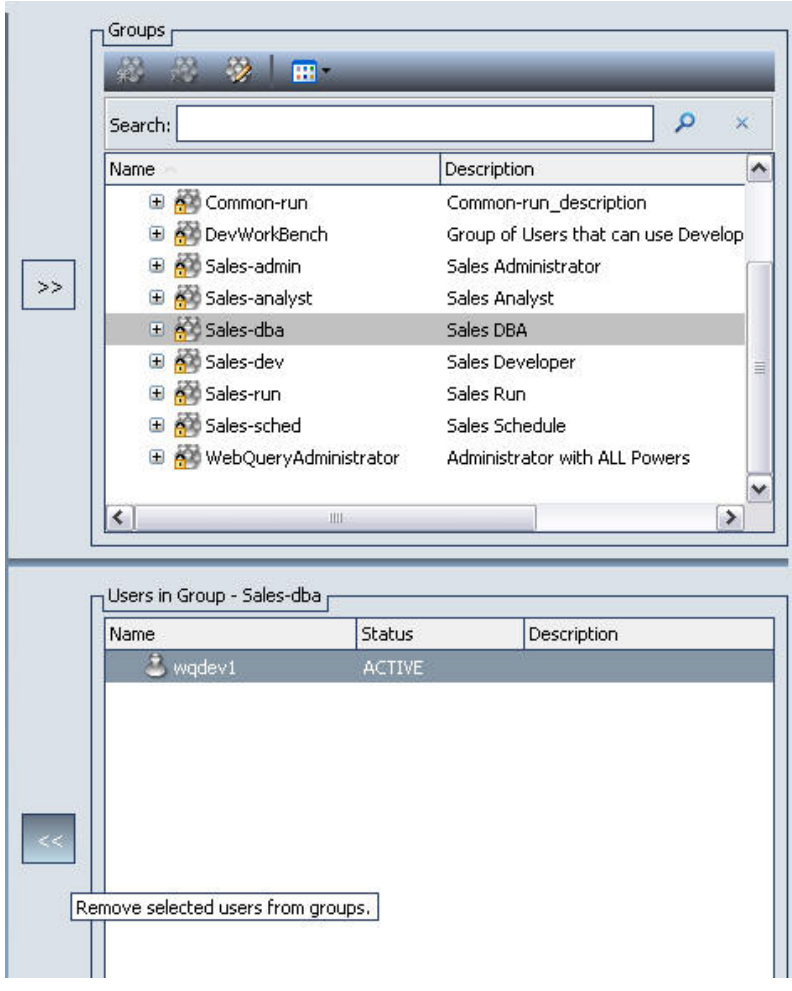
   Notice the Sales-run group icon has changed to indicate a linkage to an IBM i group profile.

9. Click *Close* to exit the Security Center.

Procedure: How to Remove a User From a Web Query Group

1. Launch the Security Center.

2. In the Groups panel, select the group that you want to modify. The members of the selected group are listed in the Users in Group panel, as shown in the following image.

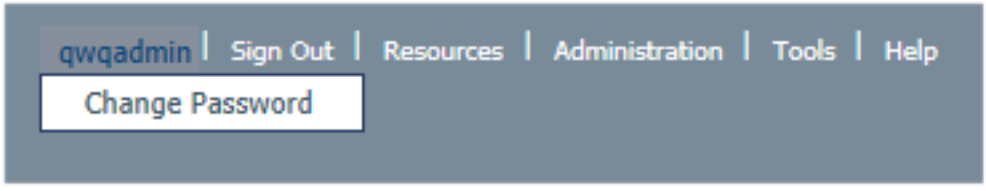


The reference to groups in this context do not refer to IBM i group profiles. In the proceeding image, a user is being removed from the Sales-dba role and not an IBM i group profile.

3. In the Users in Group panel, select the user or users that you want to remove from the group. You can choose multiple users by pressing the Ctrl key as you make your selections.
4. Click the *Remove selected users from group* button to remove the users from that group.
5. Click *Close* to exit the Security Center.

Procedure: How to Change a User Password

1. Click the user ID description on the menu bar and select *Change Password*.



2. Type the current password, then the new password, and then type the new password again to confirm the change.

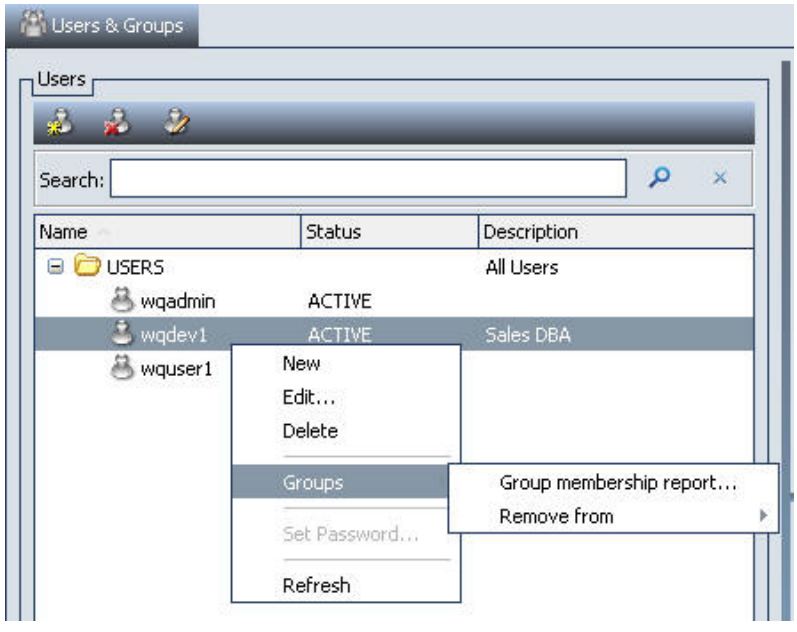
3. Click *OK* to save the changed password.

Procedure: How to Manage a User and Their Attributes

If you are an Administrator:

1. Launch the Security Center.

2. In the Users panel, right-click the user and select any one of the following options, as shown in the following image.



New

Creates a new user.

Edit

Edits the user attributes.

Delete

Deletes the user.

Groups

Opens a submenu that contains the following options:

- **Group membership report.** Generates a report listing all the groups to which the user belongs.
- **Remove from.** Opens a dialog box to remove the user from one or more groups.

Refresh

Refreshes the Users panel.

Web Query Administration

Web Query provides several administration consoles to manage the different components of Web Query. There are three consoles:

- **Administration Console.** Manages Web Query options and configuration settings, including Report Broker configuration settings.
- **Report Broker Console.** Manages the Report Broker distribution server, schedules, and blackout dates.
- **Reporting Server Console.** Manages the reporting server configuration, for example, NLS, JVM, and diagnostic settings.

Administration Console

The Administration Console is used to manage Web Query options and configuration settings, including Report Broker Distribution Server configuration settings. The Administration Console is available to the QWQADMIN user ID or any member of the WebQueryAdministrator group.

Procedure: How to Launch the Administration Console

1. Sign in to Web Query.
2. Click the *Administration* option on the menu bar and select *Administration Console*. The Web Query Administration Console opens.

   Notice that there is a Report Broker tab that is used to manage the Distribution Server Configuration settings.

   Note: Click the *Help* button for detailed online Help on the Administration Console.

Report Broker Console

The Report Broker Console is used to manage the Report Broker distribution server, jobs, log files, blackout dates, and global updates.

Reporting Server Console

Th Reporting Server Console is used to manage the Reporting Server configuration, for example, NLS, JVM, and diagnostic settings. The Reporting Server Console is available to the QWQADMIN user ID or any member of the WebQueryAdministrator group.

Procedure: How to Launch the Reporting Server Console

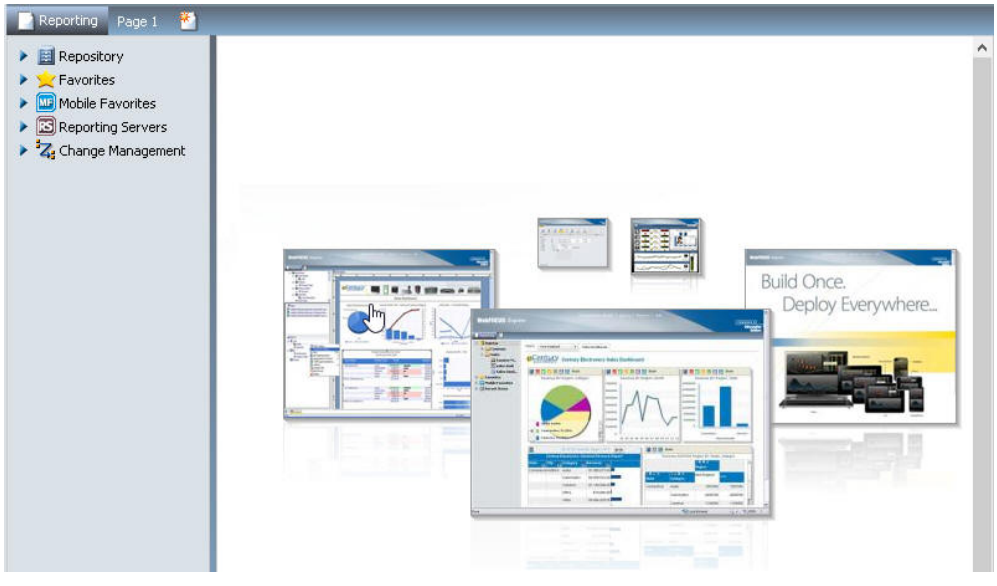
1. Sign in to Web Query.

2. Expand the Reporting Servers node.
3. Right-click *EDASERVE* and click *Reporting Server Console*. The Reporting Server Console appears.

   Note: Click the *Help* option for detailed online Help on the Reporting Server Console.

Web Query User Interface

Once you sign in to Web Query, you are presented with the Reporting page as the default view. The Reporting page displays the Repository tree in the left pane and the Output panel in the right pane, as shown in the following image. The menu bar displayed at the top of the window has commonly used commands, such as Sign Out and Help.



Describing the Repository Tree

You can access the Db2 Web Query tree, as shown in the following image. Repository is the top-level node. It contains one predefined folder, Common. The Administrator manages the content in the Common folder.

The Repository tree, located on the left side of the page, contains your Db2 Web Query Content. The following image shows the Repository tree that is displayed when you are signed in to Db2 Web Query. The content is secured using the security system in a folder-based system similar to desktop operating systems. The content will differ for every user, including the public user, who will see only public content.

In order to separate your applications, you can create folders in the Repository. For example, you can have a Payroll folder and a Sales folder to separate these two applications. You can also create and store procedures, report output, and schedules in the Repository.

The right-click options in the Repository tree allow you to launch the development tool, schedule procedures for distribution, and administer users and authorization to objects.

- **Repository.** This is where applications are developed. Its contents are folders, procedures, and scheduling objects.
- **Favorites.** The Favorites folder contains objects that have been added using the Add to Favorites option from the tree.
- **Mobile Favorites.** The Mobile Favorites folder contains procedures that have been added using the Add to Mobile Favorites option from the tree. These procedures are easily accessible from mobile devices, such as an Apple iPhone or Android phone.
- **Reporting Servers.** The Reporting Servers node appears for administrators only. Only one Reporting Server is configured with the product. This is where metadata is managed across applications.
- **Change Management.** The Change Management node links you to tools that create change management scenarios, download and export them for use in applications, and upload and import change management packages delivered to you.

Using Right-Click Options in the Repository Tree

In the Repository tree, you can right-click an object and select from a list of available options that are displayed in the context menu. Some options are available only to Administrators.

If you right-click the Repository tree, the following options are available.

View

Allows you to view the contents of the Repository.

- Display By Title.
- Display By Name.

New Folder

Creates a new top-level folder.

Refresh

Refreshes the Repository tree.

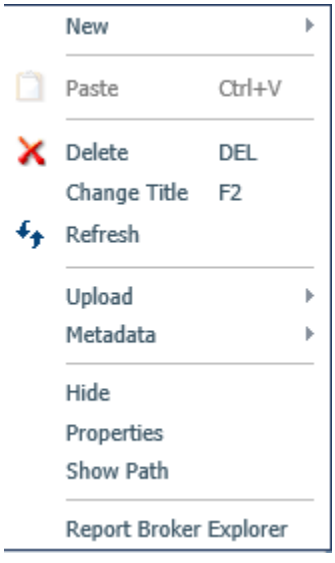
Paste

Pastes a copied item into the Repository.

Report Broker Explorer

Launches the Report Broker Explorer to manage Report Broker objects. This option is available to Schedulers and Administrators.

If you right-click a top-level folder or subfolder in the Repository, the following options are available.



New

Opens a submenu that contains the following options:

- **Report.** Launches InfoAssist+ in Report development mode.
- **Chart.** Launches InfoAssist+ in Chart development mode.
- **Document.** Launches InfoAssist+ in Document development mode.
- **Visualization.** Launches InfoAssist+ in Visualization development mode.
- **Text Editor.** Launches the text editor. This option is only available to the Web Query Administrator group and should not be used to develop procedures. It is intended for debugging only.
- **URL.** Launches a Create URL dialog box to prompt for necessary information to create a URL report.
- **Distribution List.** Launches a dialog box from which you can create a new distribution list for Report Broker.
- **Folder.** Creates a private subfolder.

Paste

Pastes a folder that was copied. This option is available only in subfolders.

Delete

Deletes the selected folder.

Change Title

Changes the description or name of the folder.

Refresh

Refreshes the contents of the selected folder.

Upload

Uploads data, documents, or images into the Repository. You can also access the Upload Wizard from this option.

Metadata

Launches the Metadata Manager to guide you through the process of defining connections and metadata that map to your data sources.

- **New.** Launches the Metadata Manager and directs you to My Console.
- **Edit.** Launches the Metadata Manager and directs you to the Applications Directories tree where synonyms can be edited to contain custom fields, descriptions, and joins.
- **Metadata Wizard.** Launches the Connect to Data wizard to configure a connection to a data source.

Hide and Show

Hides or shows a folder from a run only user. This option is only available to Developers and Administrators.

Properties

This option is only available to Web Query Administrators to view report properties. A Web Query Administrator is any user belonging to the WebQueryAdministrator group.

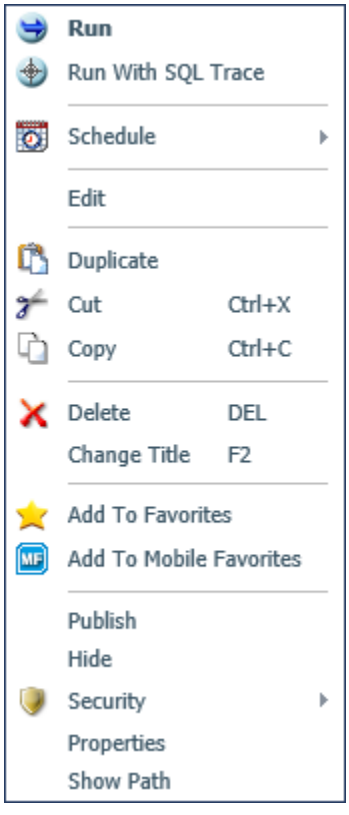
Show Path

Displays the full path of the object within the Repository.

Report Broker Explorer

Launches the Report Broker Explorer to manage Report Broker objects. This option is available to Schedulers and Administrators.

If you right-click a report in the Repository, the following options are available.



Run

Runs the selected report in the report output panel.

Run With SQL Trace

Runs the report with SQL Trace. The SQL Trace will appear in a panel below the report.

Schedule

Sets the distribution method for the schedule.

- **Email.** Distributes the report through email.
- **FTP.** Distributes the report through FTP.
- **Printer.** Distributes the report to one or more printers.
- **Repository.** Distributes the report back to the Repository.

Edit

Opens the selected report using InfoAssist+.

Edit With

Opens a submenu allowing you to edit the report using InfoAssist+ or a text editor.

Duplicate

Duplicates the selected report.

Cut

Cuts the selected report.

Copy

Copies the selected report.

Delete

Deletes the selected report.

Change Title

Changes the title of the report.

Add To Favorites

Adds the selected report to the Favorites folder in the Repository.

Add To Mobile Favorites

Adds the selected report to the Mobile Favorites folder in the Repository.

Publish and Unpublish

Publishes or unpublishes the report to allow public access to users who belong to the folder.

Hide

Hides a folder from being displayed run only user. This option is only available to Developers and Administrators.

Show

Displays a folder for a run only user repository.

Security

Opens a submenu that allows you to set the Resource owner for the selected report. Owner opens the Set Owner dialog box.

Properties

This option is only available to Web Query Administrators to view report properties. A Web Query Administrator is any user belonging to the WebQueryAdministrator group.

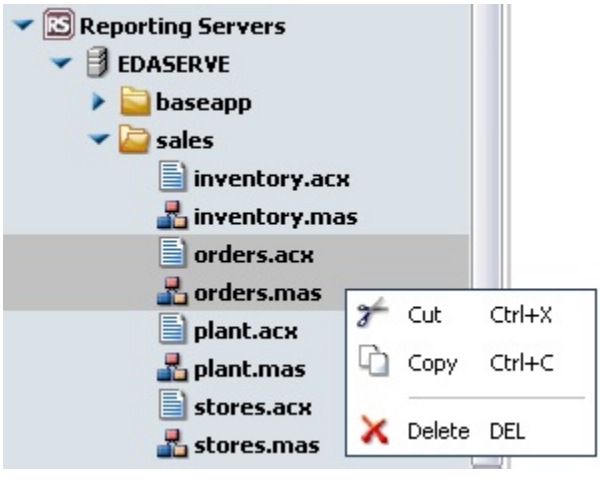
Show Path

Displays the full path of the object within the Repository.

Administering the Reporting Server

The Reporting Servers node appears for Administrators only. This allows an administrator to manage, delete, copy, and paste synonyms across application folders. This functionality is needed when a procedure is copied to a different folder. The synonym or synonyms used by the procedure must also be copied to that same folder, in order for the report to run in the new folder.

The baseapp folder is visible to all users. A synonym that needs to be shared by all users should be copied into the baseapp folder.



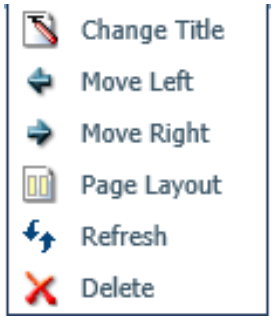
Creating a Page

You can create multiple pages that show different reports. To create a page, click the New Page  icon located above the tree.

Press the *F8* function key or click the *Resources* link on the menu bar to open the Resource Tree, as shown in the following image.

You can drag reports from the Resource Tree onto the blank page.

You can right-click any page tab and select from a list of available options in the context menu, as shown in the following image.



Change Title

Renames the selected page.

Move Left

Moves the selected page to the left.

Move Right

Moves the selected page to the right.

Page Layout

Chooses a page layout to align your content into the desired number of columns.

Refresh

Refreshes the contents of the page by running all procedures on the page.

Delete

Deletes the selected page.

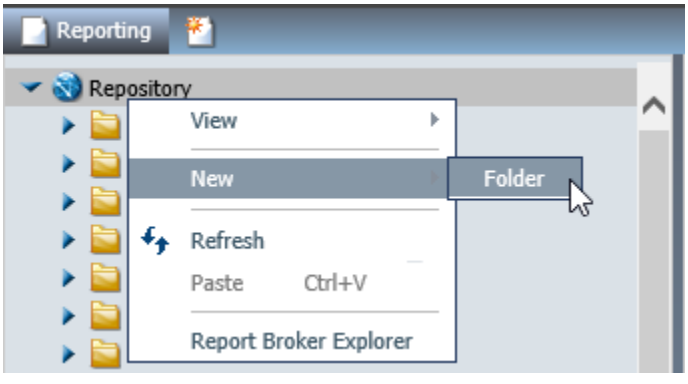
Managing Folders

The Repository is composed of folders and subfolders. A top-level folder that is created in the Repository defines applications for a specific set of users. Users are associated with a folder by adding them to one or more Groups that are created with each top-level folder.

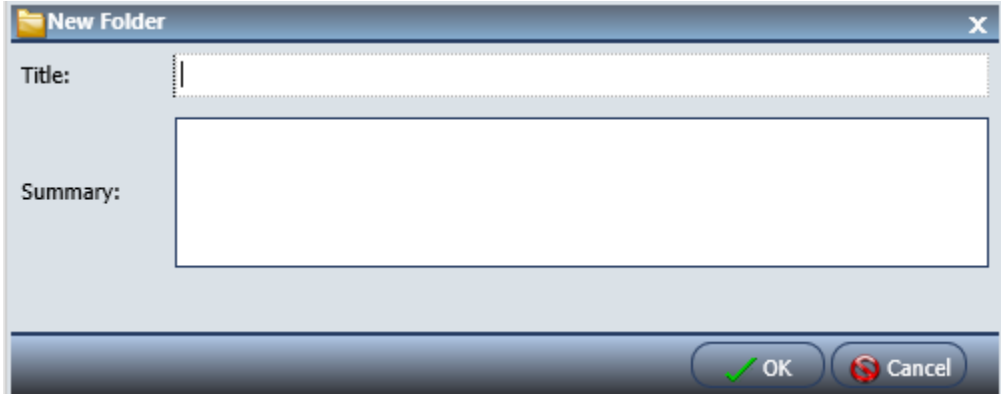
A top-level folder is created as a Published Object by default. A subfolder is created as a Private folder and only the owner of the folder can access objects within it. The owner of a Private folder can change the permissions of the folder by right-clicking the folder and selecting *Security*.

Procedure: How to Create a Folder

1. Right-click *Repository* and click *New Folder*, as shown in the following image.



The New Folder dialog box opens, as shown in the following image.



2. Type a title and summary in the provided input fields. By default, the folder description is displayed in the Repository tree.

   Note: If you create a folder with the same title, Web Query will automatically create a unique name for the folder. To see a folder name, right-click the folder and click *Properties*.

3. Click *OK* to create the folder.

Selecting a Top-Level Folder

All of the top-level folders that a user is authorized to access are displayed in the Repository tree. Depending on how your user profile is set up, you may not have access to all top-level folders.

The icon located next to an item in the top-level folder represents the type of item.

Building a Report, Chart, Document, or Visualization

InfoAssist+ is a single development tool used to create reports, charts, documents, and visualizations. There are four modes in which the tool can be launched.

Procedure: How to Build a Report, Chart, Document, or Visualization

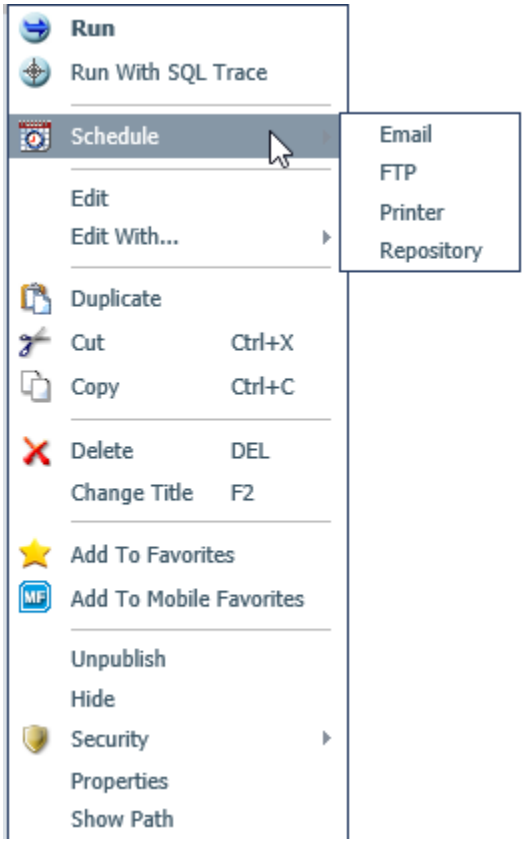
1. Right-click a folder in Web Query and select *New*.
2. Select the mode that is required:
   - **Report.** Launches InfoAssist+ in Report development mode.
   - **Chart.** Launches InfoAssist+ in Chart development mode.
   - **Document.** Launches InfoAssist+ in Document development mode (combination of reports and charts).
   - **Visualization.** Launches InfoAssist+ in Visualization development mode.

Scheduling a Procedure

You can schedule to run any procedure that you develop using InfoAssist+ and electronically distribute the output using Report Broker.

Right-click a procedure from the Repository tree and select *Schedule*, then the type of distribution method. The supported distribution methods are Email, FTP, Printer, and Repository.

The Repository distribution method will deliver your report output into the same folder where the procedure resides.



Using Menu Bar Hyperlinks

The hyperlinks in the menu bar provide a way to easily navigate the reporting interface:

- **Sign-in name.** Allows you to change the password or remove customizations.
- **Sign Out.** Allows you to sign out of the interface.
- **Resources.** Contains your Web Query content.
- **Administration.**
  - Security Center
  - Administration Console (if signed in as administrator)

- Manage Private Resources
- Switch between Normal mode (seeing your own content) and Manager mode (which allows you to manage other user content)

- **Tools.**
  - **FOC Message Lookup.** Lookup detailed error message text for a given error code.
  - **Stop Requests.** Terminate a running Web Query procedure.
  - **Session Viewer.** Display information about your current work session. It also lists entries for all recently completed sessions. This option is available for Web Query administrators, developers, and analysts.
  - **Runtime Environments.** Dynamically change library lists. For more information, see the Db2 Web Query Dynamic Runtime Implementation guide at:

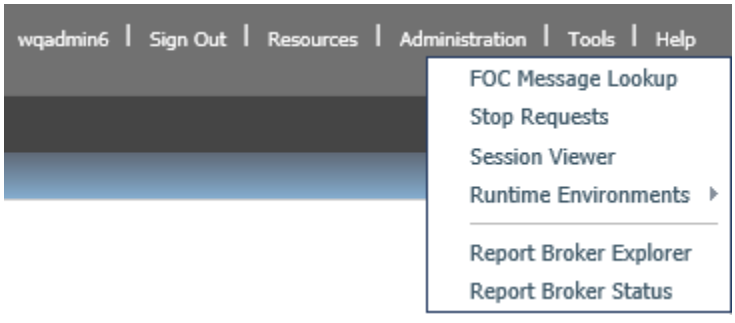
    https://www.ibm.com/developerworks/mydeveloperworks/wikis/home?lang=en_US#/wiki/W516d8b60d32c_4fc5_a811_5f3d840bf524/page/Dynamic%20Runtime%20Environments
  - **Report Broker Explorer.** Launch this tool to manage Report Broker schedules and distribution lists. Refer to the Report Broker online help for more information on the Report Broker tools.
  - **Report Broker Status.** Launch this tool to manage the Report Broker configuration. Refer to the Report Broker online help for more information on the Report Broker tools.
- **Help.** Opens a menu where you can select:
  - *Db2 Web Query Online Help* which opens the online help.
  - *About Db2 Web Query* which provides release information.

Understanding Db2 Web Query Error Text

You can use the Error Message Lookup feature to understand what an error messages means. The link provides the full text associated with a Reporting Server error message (FOCXXXX). To use this feature, you need the numeric portion of the error message. Type only the numeric portion of the error number. If you enter "FOC", an error will occur. This option is available for administrators and developers only. Run only users will not see this option. To access the Error Message Lookup feature, click the *Tools* link located on the Db2 Web Query banner.

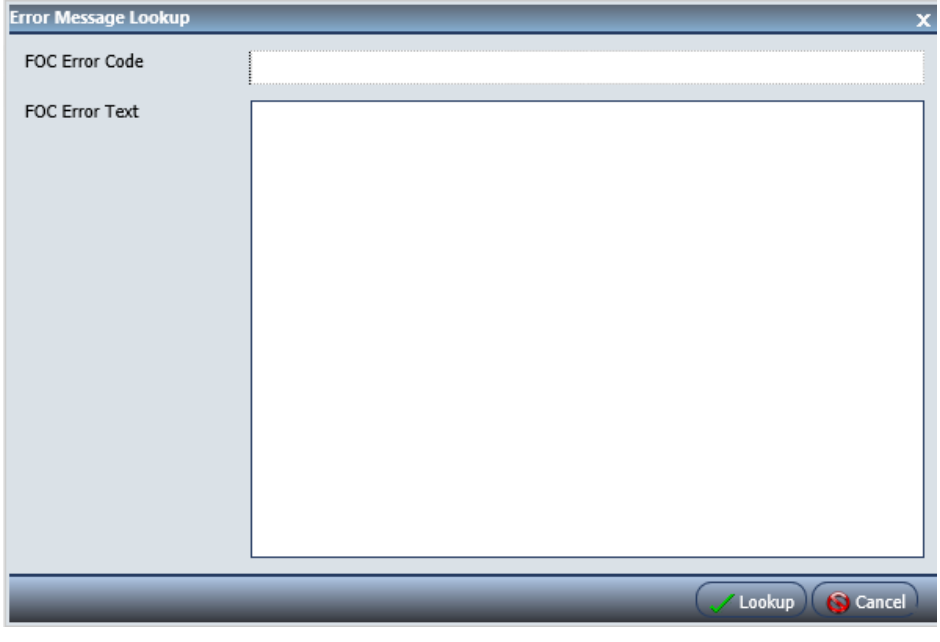
Procedure: How to Define an Error Message

For administrators and developers, the banner contains the Tools link, as shown in the following image.



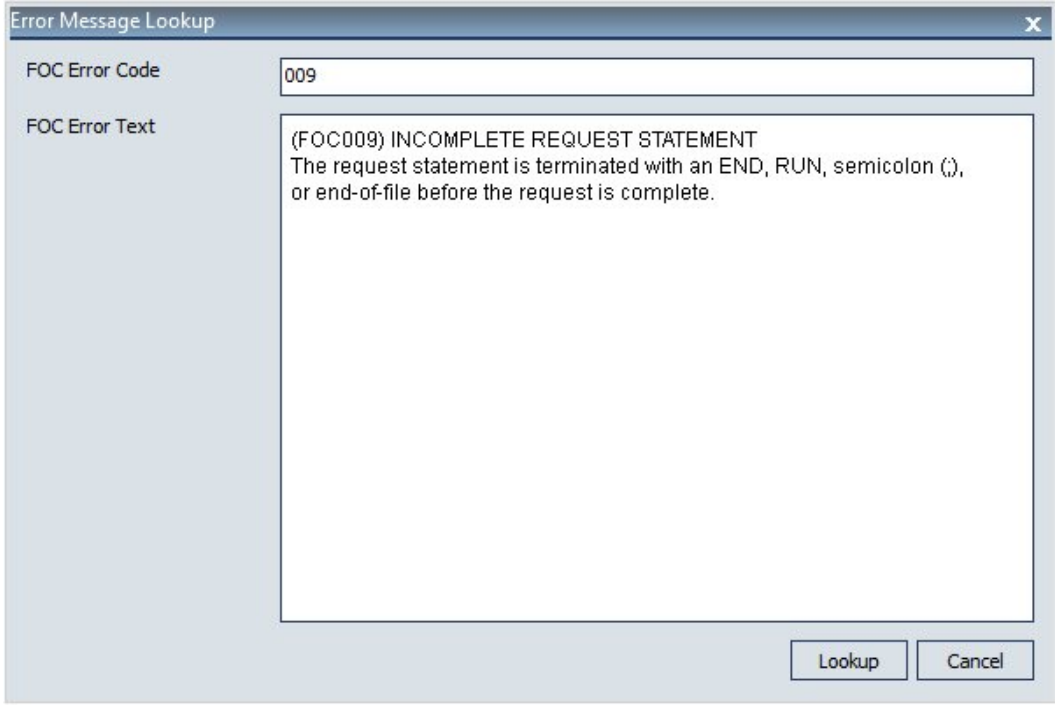
1. Select *FOC Message Lookup*.

   The Error Message Lookup screen appears, as shown in the following image.



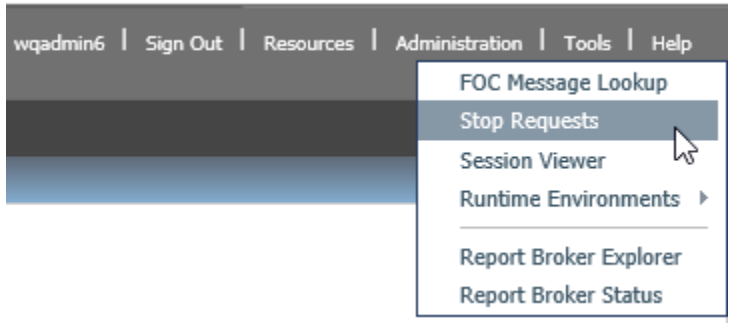
2. Type the numeric portion of the error message in the FOC Error Code field.
3. Click *Lookup*.

The full text description associated with the FOC error message appears, as shown in the following image.



Stopping a Db2 Web Query Request

Users can cancel active requests that are initiated from their browser session on the Reporting Server by clicking *Stop Requests* on the top banner, as shown in the following image.



Reference: Stopping a Db2 Web Query Request

All active requests for the current browser session are canceled and the following confirmation message displays:

```
xx request(s) stopped.
```

where:

xx
: Is the total number of requests that were canceled.

  Note: The Reporting Server may take some time to complete the stop request. Clicking *Stop Requests* during this time may result in an incorrect number of canceled requests being reported.

Once the request is canceled, Db2 Web Query displays the following message to the user that ran the request:

```
Reporting server request terminated by operator.
```

If the request is stopped while data is already being output, the following message displays in the report output:

```
This report is invalid because the data retrieval has been killed or the
job has been stopped.
```

If the report output is a PDF and the request is stopped while data is already being output, the following message displays:

```
The File is damaged and could not be repaired.
```

Chapter 3

Using the Db2 Web Query Spreadsheet Client Add-in

The Spreadsheet Client is a Microsoft Office add-in that enables you to connect Excel directly to Db2 Web Query reporting tools, where you can access and analyze data. Connecting Excel to the Db2 Web Query reporting engine allows Spreadsheet Client to leverage all the adapters available to Db2 Web Query.

You can install the Spreadsheet Client Add-in on your desktop, where you can create and edit queries by accessing predefined data sources. Queries can be saved in an Excel document and refreshed at any time.

Because all Db2 Web Query report and connection information can be saved in an Excel workbook, users with the proper security and access rights can share spreadsheets throughout an organization. This functionality enables you to spend less time recreating reports and more time analyzing information for effective decision-making. It also enhances Excel data-privacy features and better addresses compliance concerns. When reports are built directly within Excel, you can lock all or some of the cells and password protect the worksheet. Locking cells can be useful to protect data and queries when sharing workbooks, but in Excel, locked cells cannot be refreshed or edited.

In this chapter:

Spreadsheet Client Features

When using Spreadsheet Client from within the familiar Excel environment, you can utilize many powerful features including, but not limited to, the following:

- Build Excel applications with real-time information fed by Spreadsheet Client.
- Create dashboards and scorecards in Excel by combining multiple data sources in a single worksheet.
- Ensure data integrity in your spreadsheets by using named ranges and formulas that are automatically generated and updated with Spreadsheet Client.
- Create queries from scratch using the simple, yet powerful ad hoc reporting capabilities of Db2 Web Query.
- Use your own custom guided ad hoc forms to populate the workbook.
- Output computations and totals as native Excel formulas.
- Style output, add data filtering, and include drill-downs.
- Supports 64-bit version of Excel 2010.
- Supports 32-bit version of Excel 2003, 2007, and 2010.

Configuring Spreadsheet Client

The Db2 Web Query Administration Console is used to configure the Spreadsheet Client. By default, the Spreadsheet Client is configured to use Db2 Web Query Managed Reporting Security. This leverages the security defined in Db2 Web Query and provides secure access to all data available through the Db2 Web Query environment. Additional optional settings in the Console can be configured to provide you with the ability to create queries by accessing SAFs (Structured Ad hoc Forms) stored in the Managed Reporting repository. A SAF is an HTML form containing a report procedure that is already connected to a data source, which enables you to select from a series of parameters to build a data set for analysis in Excel.

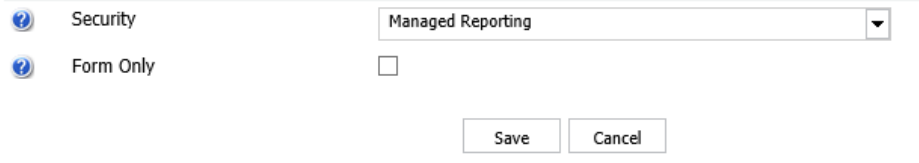
Note: Run-time enablement users are not permitted to use InfoAssist+ and must use SAFs.

Spreadsheet Client has been enhanced when using Managed Reporting security to ensure that users are presented with a list of Master Files based on the application path set in the properties of a repository folder. This ensures that Managed Reporting Security is properly utilized and users only have access to metadata for which they are authorized.

Procedure: How to Configure Spreadsheet Client

1. Sign in to Db2 Web Query using a Web Query Administrator user ID, and launch the Administration Console. For more information, see *Administration Console* on page 26.
2. Click *Configuration* in the navigation pane, then click *Spreadsheet Client*.

   The Db2 Web Query Spreadsheet Client panel appears with the *Security* setting set to Managed Reporting, by default, as shown in the following image. Do not change this setting.



3. The *Form_Only* parameter is not selected, by default. If you select the check box, users can access only the available Structured Ad Hoc Forms (SAFs). Users will not be able to use InfoAssist+ to create a report.
4. Click *Save* to save your configuration settings. Click *Clear Cache* on the main toolbar to clear the site collection cache for the web application.

Configuring a Default Db2 Web Query Environment

A configuration file is provided with the Spreadsheet Client Add-In as a template for the administrator to design a default Db2 Web Query environment. The configuration file defines such items as the Db2 Web Query Web server port number, alias, and client path. These items provide the user with a default Db2 Web Query environment allowing them to bypass the additional step of manually defining these parameters in order to use Spreadsheet Client.

The configuration file is named wqsclient.cfg and is located in

`/QIBM/ProdData/QWEBQRY/base80/utilities/quickdata`

The configuration file can contain multiple Db2 Web Query configurations. Keep in mind that if the configuration file contains more than one Db2 Web Query configuration, then the last one appearing in the file is the configuration that is used when opening Spreadsheet Client.

The configuration file provided as a template with the Spreadsheet Client Add-in contains examples of configurations and instructions to help you create your own configuration. The following is an example of a Db2 Web Query configuration in the configuration file:

```
SERVER_START
     PROTOCOL="http"
     HOST="wq_hostname"
     PORT="12331"
     HTML_ALIAS="/webquery_html"
     CLIENT_PATH="/webquery/WFServlet"
SERVER_END
```

Use the following guidelines and rules to create the configuration file:

- The configuration file must have the same name as the Spreadsheet Add-In file and the extension .cfg (for example, wqsclient.cfg).
- The configuration file must reside on the machine running the Spreadsheet Client in the same directory as the .xla file.
- Each Db2 Web Query configuration must be contained by the delimiters, SERVER_START and SERVER_END.
- Each configuration must contain the following parameters in order to connect to Db2 Web Query:
  - **PROTOCOL.** The protocol used in the environment running Db2 Web Query. If Db2 Web Query is running in an SSL environment, you must specify https as the protocol value. The default value is http.
  - **HOST.** The server name where the Db2 Web Query Web application is installed.
  - **PORT.** The port number of the application server where Db2 Web Query is installed. The default port is 12331. This should not be changed.
  - **HTML_ALIAS.** The Web server or application server alias where the Db2 Web Query static pages are located. The default value is /webquery_html and should not be changed. The leading slash is required.
  - **CLIENT_PATH.** The path to the Db2 Web Query Servlet, as defined in the Db2 Web Query Web application file, web.xml. This value is /webquery/WFServlet and should not be changed. The leading slash is required.
  - Using double quotation marks (" ") around parameter values, as shown in our example, is optional.
  - Begin a comment line in the file with a number sign (#).

Installing and Setting Up the Spreadsheet Client

The Db2 Web Query Spreadsheet Client requires:

- Db2 Web Query Version 2.1.x or higher.
- Excel 2003 or higher.

Procedure: How to Distribute the Web Query Spreadsheet Client Add-in File

1. Copy the wqsclient.xla add-in file and wqsclient.cfg configuration file, located in the following directory:

   `/QIBM/ProdData/QWEBQRY/base80/utilities/quickdata`

2. Add the two files to the following directory on the machines that use the Web Query Spreadsheet Client:

   drive`:\Users\`*userid*`\AppData\Roaming\Microsoft\AddIns\`

   where:

   userid

   Is the user name used to sign in to the PC.

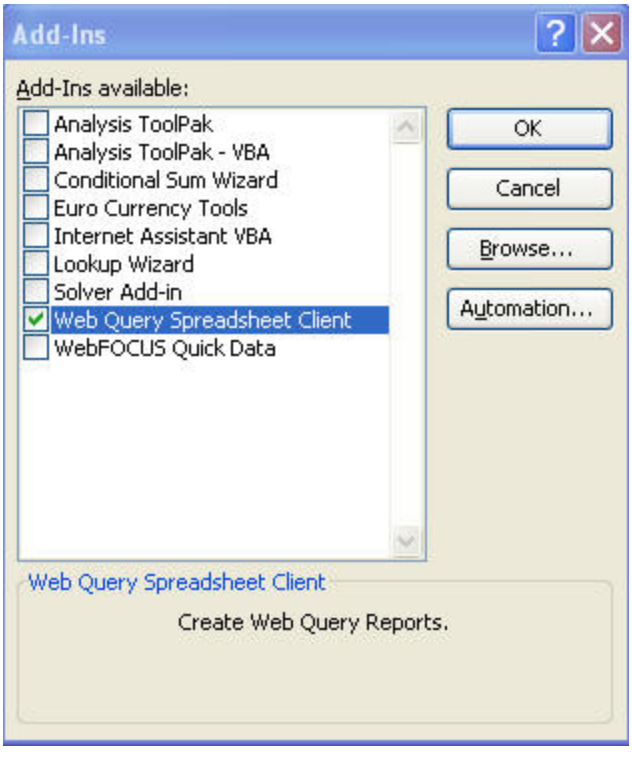
   After the Spreadsheet Client Add-in is in the proper directory location, you must open Excel and select the Db2 Web Query Spreadsheet Client option in the Add-Ins dialog box.

Procedure: How to Enable the Web Query Spreadsheet Client in Excel 2003

Use the following steps to enable the Web Query Spreadsheet Client in Excel 2003.

1. Launch Microsoft Excel.
2. Click *Tools*, and then select *Add-Ins*.

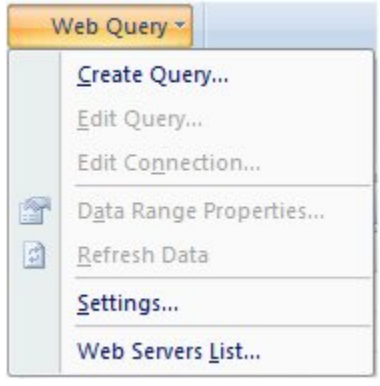
The Add-Ins dialog box appears with Db2 Web Query Spreadsheet Client listed as a selectable add-in option, as shown in the following image.



Note: If Db2 Web Query Spreadsheet Client is not listed in the Add-Ins dialog box, check that the add-in is in the correct directory.

3. Select *Web Query Spreadsheet Client* and click *OK*.
4. Close and open Microsoft Excel.

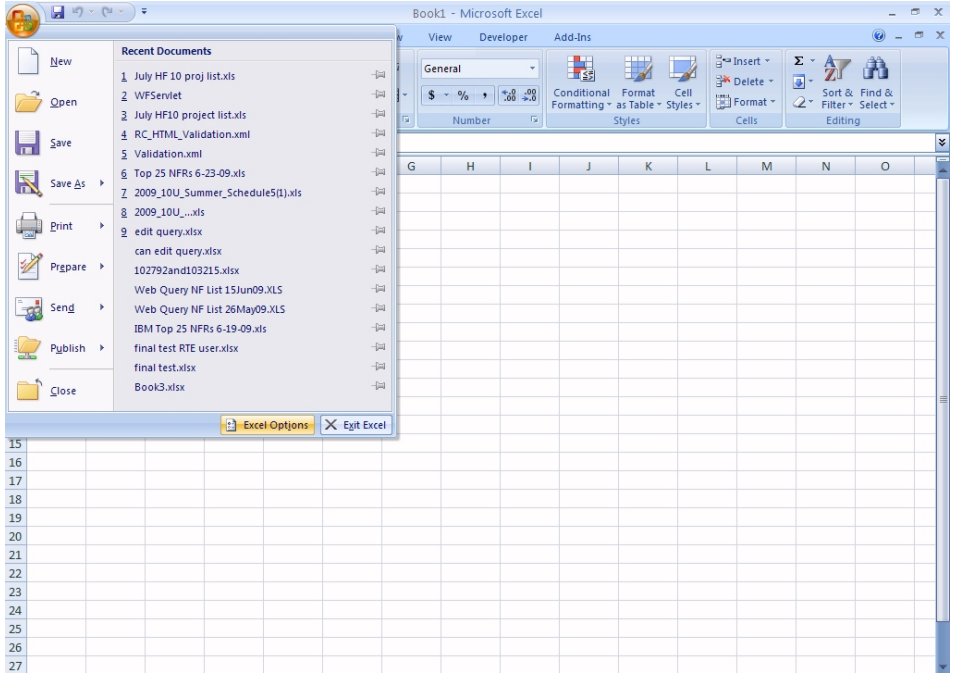
A new menu, labeled Web Query, is enabled in Excel, as shown in the following image.



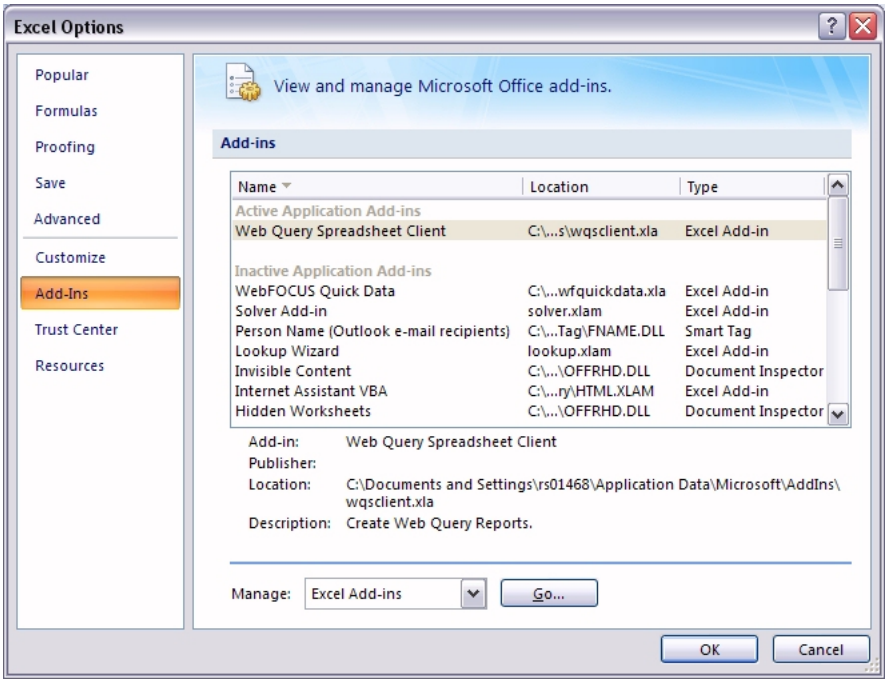
Procedure: How to Enable the Web Query Spreadsheet Client for Excel 2007

Use the following steps to enable the Web Query Spreadsheet Client in Excel 2007.

1. Launch Microsoft Excel.
2. Click the *Office* button.
3. Select *Excel Options*, as shown in the following image.



4. Click the *Add-Ins* option.
5. Click the *Go* button to the right of the Excel Add-in drop-down mwnu, as shown in the following image.



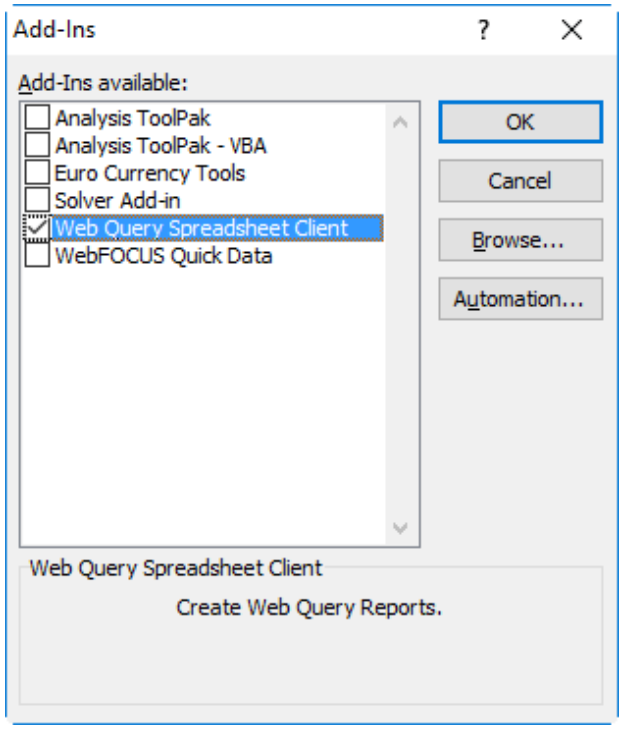
   The Add-Ins dialog box appears.
6. Select *Web Query Spreadsheet Client* and click *OK*.

Procedure: How to Enable the Web Query Spreadsheet Client in Excel 2010 and Higher

Use the following steps to enable the Web Query Spreadsheet Client in Excel 2010 and higher.

1. Launch Microsoft Excel.
2. Click the *File* tab, click *Options*, and then click *Open*.
3. In the Excel Options dialog box, click *Add-Ins*.
4. In the Manage dialog box, click *Excel Add-ins*, and then click *Go*.

The Add-Ins dialog box opens, with Web Query Spreadsheet Client as an add-in option, as shown in the following image.



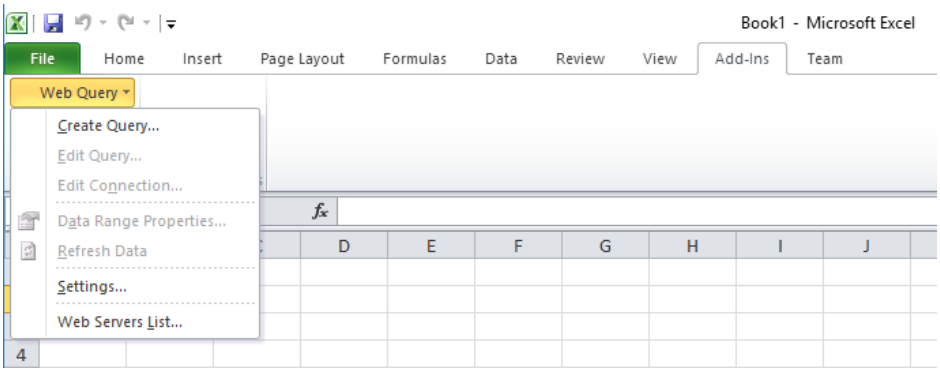
Note:

- If Web Query Spreadsheet Client is not listed in the Add-Ins dialog box, ensure that the add-in file is installed and placed in the correct directory.
- If the Web Query Spreadsheet Client add-in was installed in a different directory, use the *Browse* button in the Add-Ins dialog box to locate it.

5. Select the *Web Query Spreadsheet Client* check box and click *OK*.

   The Add-Ins dialog box closes, and the Add-Ins tab appears on the ribbon.

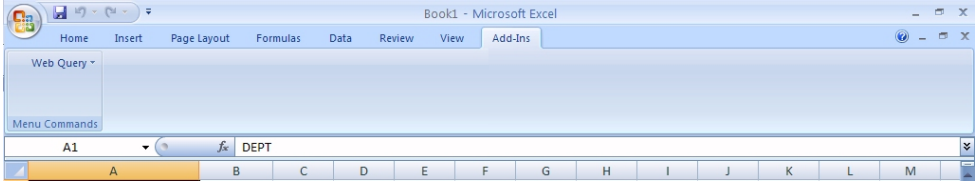
6. Click the *Add-Ins* tab to view the new Web Query menu, as shown in the following image.



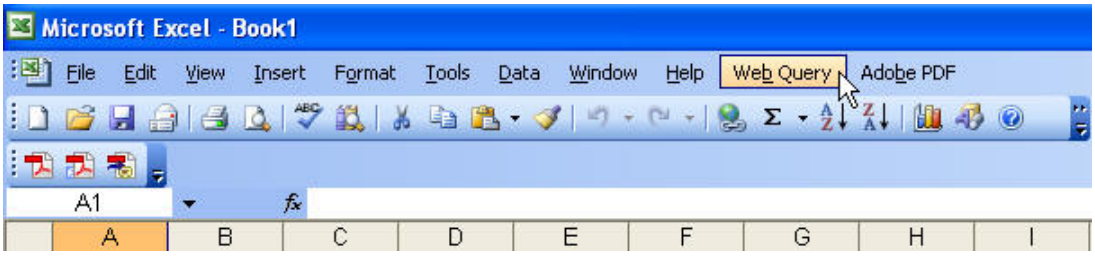
Accessing Spreadsheet Client

After installation, Spreadsheet Client is available in Excel from the main tool bar.

In Excel 2007, the Add-ins tab is displayed on the main tool bar and contains all add-in menu items, including the Web Query add-in, as shown in the following image.



In Excel 2003, the main tool bar contains the Web Query add-in as a new menu item, as shown in the following image.

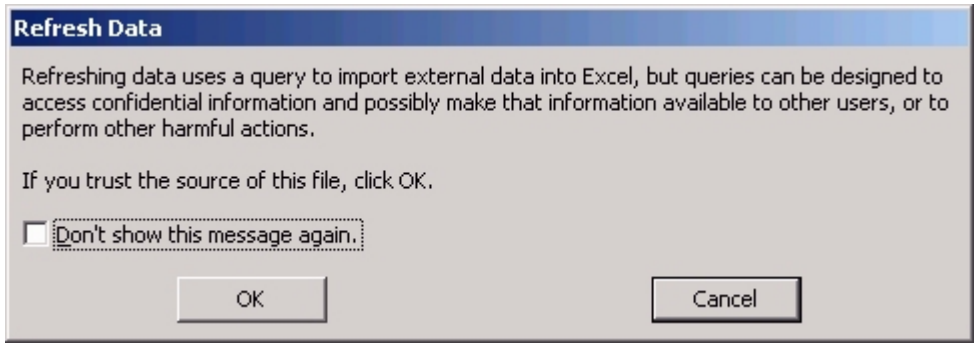


From Excel, click *Web Query* to view a drop-down menu with all of the necessary options for working with queries. The following options, except for Settings, are also available from Excel right-click context menus in cells containing any part of a query.

- **Create Query.** Available for new queries only, this option opens the Web Server Connection dialog box so you can connect to a Reporting Server. It continues by opening the Data Source Selection dialog box so you can select a Master File, and then opens InfoAssist+ where you can create the query.
- **Edit Query.** Available for existing queries only, this option opens InfoAssist+ where you can edit the query.

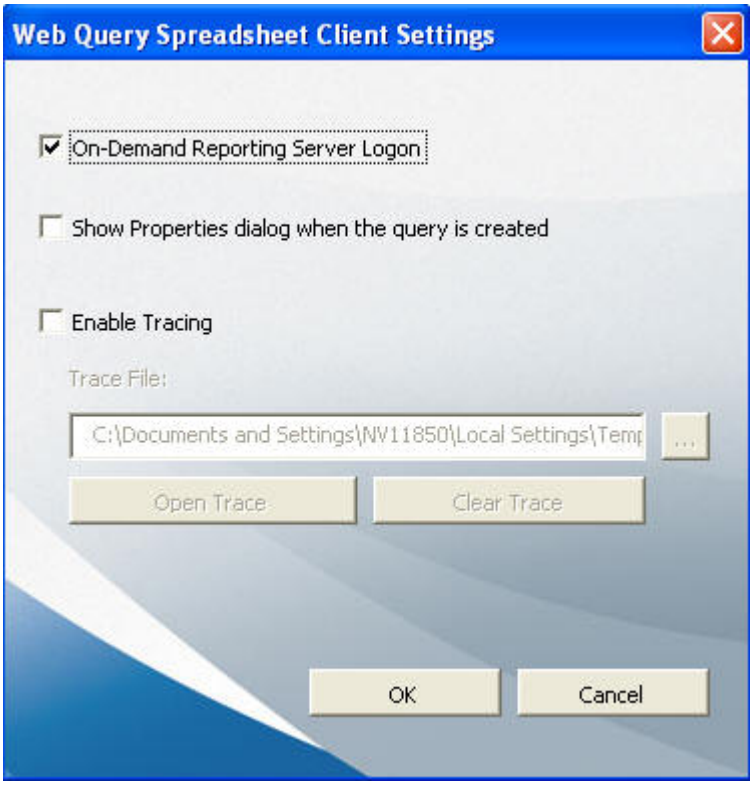
  Note: Edit Query is not enabled for password protected cells.
- **Edit Connection.** Available for existing queries only, this option opens the Web Server Connection dialog box, where you can edit the connection settings, including the Web Server URL, the HTML Alias, the Client Path, and the Reporting Server. The ability to edit connection information saves time when reusing reports and helps facilitate the sharing of workbooks across an organization.
- **Data Range Properties.** Available for existing queries only, this option opens the External Data Range Properties dialog box, where you can set Excel query properties. For more information, see *Setting Query Properties* on page 76.
- **Refresh Data.** Available for existing queries only, this option opens the Refresh Data dialog box where you can update the data in the report query.

  The Refresh Data dialog box provides a security warning, as shown in the following image.



  Note: Refresh is not enabled for password protected cells.

- **Settings.** This option opens the Web Query Spreadsheet Client Settings dialog box, as shown in the following image.



The Web Query Spreadsheet Client Settings window provides the following settings:

- **On-Demand Reporting Server Logon.** This setting determines if the user will be prompted to sign in to Db2 Web Query the first time a connection to the server is made during an Excel session (check this setting), or each time a request is made to Db2 Web Query during an Excel session (do not check this setting), for example, during a refresh or Edit Query operation.

- **Show Properties dialog when the query is created.** When this setting is selected, a dialog box with options on how to insert data into Excel opens each time a new query is executed. The following image shows this Properties dialog box.

The options to control how the results from the query will be inserted into the Excel worksheet are:

- **Insert cells for new data.** This option allows you to retain results from multiple queries in the same Excel worksheet. Depending on where new data will be inserted, data from an earlier request may shift.
- **Overwrite existing cells with new data.** This option will overwrite existing data to place results with those from the new request.

Once a query is generated, users can also control how data from new reports will be displayed in the Excel worksheet by right-clicking on a cell from an existing query and selecting *Data Range Properties*. For more information on this dialog box, see *Setting Query Properties* on page 76.

- **Enable Tracing.** This option allows you to capture Db2 Web Query Spreadsheet Client information in a trace file to troubleshoot communication problems and issues that occur when attempting to create and run report requests.

The captured information includes tasks performed by the tool when it attempts to connect to the Web Server and Reporting Server, when requests are made for data, and when data is retrieved. The default name of the trace file is wqsclient.txt and it is created in the same directory as the Db2 Web Query Spreadsheet Client add-in file, for example:

`C:\Users\`*`userid`*`\AppData\Roaming\Microsoft\Addins\`

Note: Traces are captured for the duration of a single active Excel session. Tracing is automatically turned off when you close an Excel session. The trace file content is cumulative, adding trace information from each session where tracing is enabled.

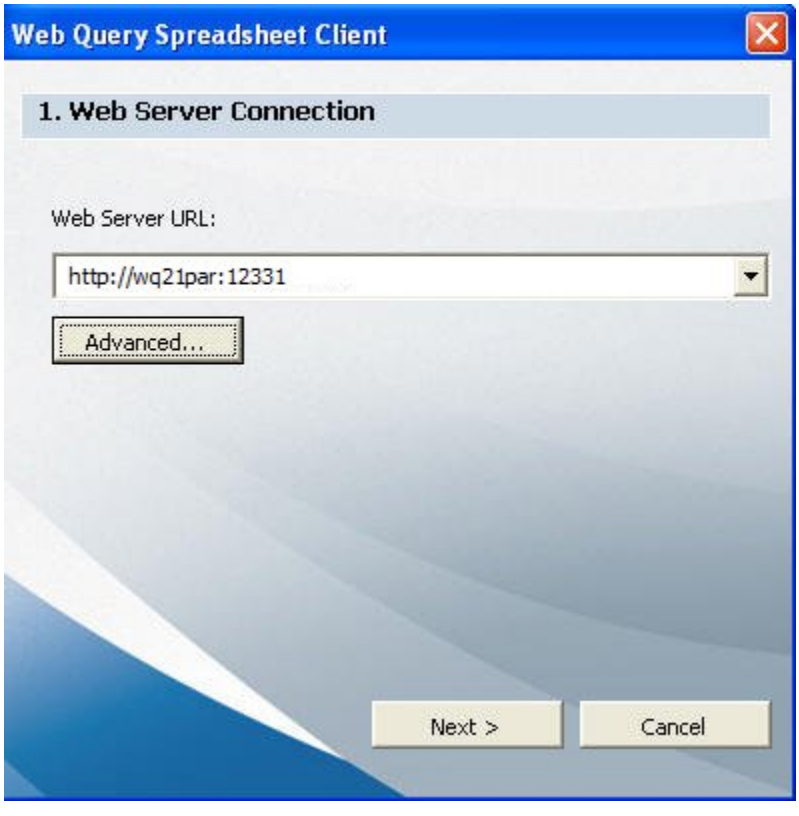
When you select *Enable Trace*, the Trace File field is automatically populated with the full path to the trace file. The path includes the trace file name. You can change the location and name of the trace file by either typing the changes in this field or by clicking the ellipsis and browsing to a new trace file location.

To view the current trace file, click *Open Trace*.

To delete the contents of the current trace file, click *Clear Trace*.

Defining Web Server Connection Settings

Building, running, and accessing a query requires an HTTP connection to a Db2 Web Query reporting environment. The Web Server Connection dialog box opens when a new query is created, as shown in the following image.



The Spreadsheet Client configuration file, wqsclient.cfg, should have been updated by your administrator to include the correct Web Server URL for your environment. For more information, see *Configuring a Default Db2 Web Query Environment* on page 45. A newly created query will use this connection, by default.

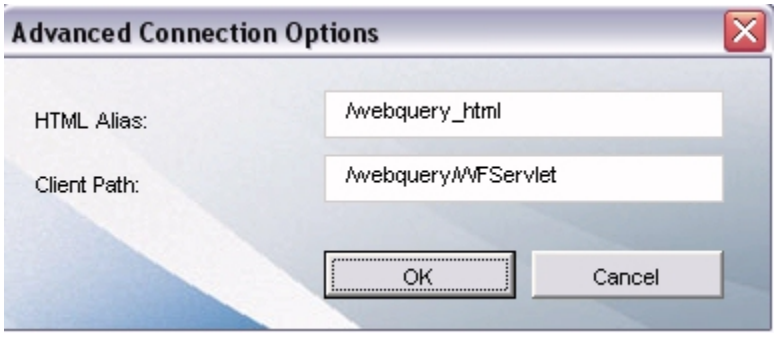
Reference: Web Server Connection Usage Notes for Queries

For query data connections, the following are supported:

- Anonymous Web Servers.
- HTTP and HTTPS.

Advanced Connection Options Dialog Box

To access the Advanced Connection Options dialog box, click the *Advanced* button in the Web Server Connection dialog box. The Spreadsheet Client configuration file contains the proper settings for your Db2 Web Query environment. Do not change these settings. The Advanced Connection Options dialog box is shown in the following image.



- **HTML Alias.** Defines the alias to the Web Server where the webquery_html directory is located.
- **Client Path.** Specifies how calls are made to the Web Server. Your Db2 Web Query environment uses the Web Query Servlet with the webquery context path, which results in the client path being set to /webquery/WFServlet, by default.

Logging into Db2 Web Query

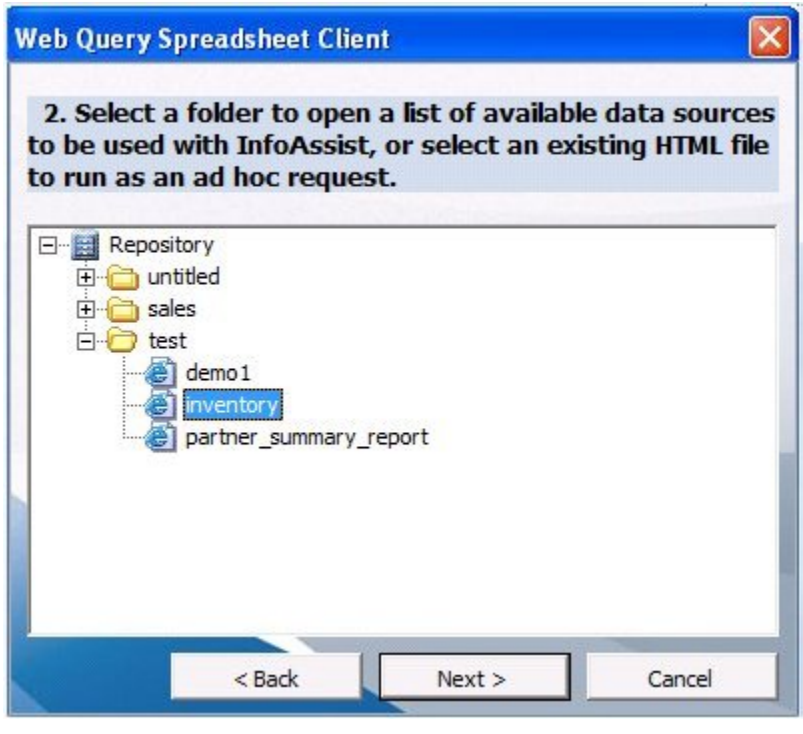
Spreadsheet Client is configured to use Db2 Web Query Authentication. You are prompted to sign in with valid Db2 Web Query Managed Reporting credentials, as shown in the following image.

If your environment is not properly licensed, an error message appears.

After a successful sign in, you will be prompted to perform one of two options:

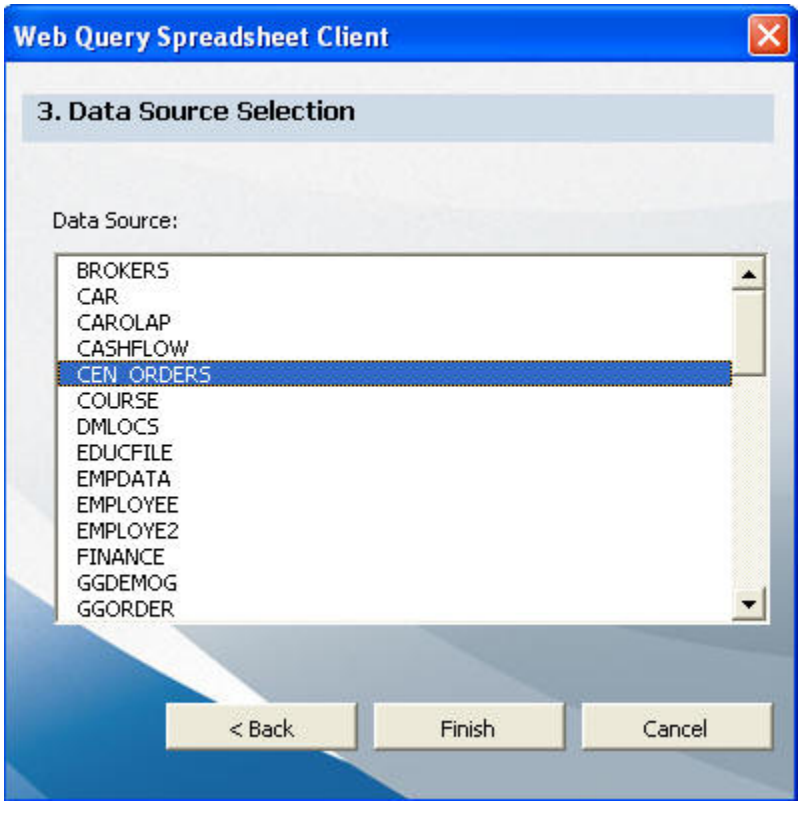
- Select a folder to determine which Master Files (synonyms) may be used to create a query with InfoAssist+. The InfoAssist+ query will then be used to populate your Excel worksheet.
- Select an existing HTML file (SAF) that will be run to populate your Excel worksheet.

Folders and HTML files will be listed based on user privileges, as shown in the following image.



Selecting a Master File

After you select a folder from the Web Query repository, you are presented with a corresponding list of Master Files (synonyms), as shown in the following image.



Scroll through the Data Source list and select the desired Master File. When you click *Finish*, the associated data fields are loaded into InfoAssist+.

Note: The only time you can select a Master File is when you are creating a new query. The Data Source Selection dialog box is not available for you to change Master Files when editing an existing query.

Creating Report Queries With InfoAssist+

You can create a new report query directly from Excel by accessing the Spreadsheet Client Add-in. Multiple queries can be placed within the same worksheet or spread out over multiple worksheets within a workbook.

Note that there are limitations with queries that overlap. However, there are data layout options available in the Query properties of Excel that can assist with overlapping queries. This behavior is governed by Excel, not Spreadsheet Client.

Procedure: How to Create a New Report Query in InfoAssist+

1. Open an Excel file.
2. Select a cell in which to place the query.
3. Click the *Web Query* option in the Excel menu, then select *Create Query*.

   You can also right-click any cell and select *Create Web Query Report*.
4. Specify the desired Web Server URL when the Web Server Connection dialog box opens.

   When the desired connection settings have been specified, click *Next*.

   You are prompted to sign in with valid Db2 Web Query credentials unless you are already signed in and have the on-demand Reporting Server Logon option checked under Web Query settings.
5. Select a folder to determine which Master Files (synonyms) may be used to create a query with InfoAssist+.
6. In the Data Source Selection dialog box, select the desired Master File and click *Finish*. For more information, see *Selecting a Master File* on page 60.

   You are presented with InfoAssist+, where you can build a query and run it to return the output data to Excel.

Example: Creating a New Report Query in InfoAssist+

This example covers multiple aspects of creating a new report query using Spreadsheet Client from within an Excel file.

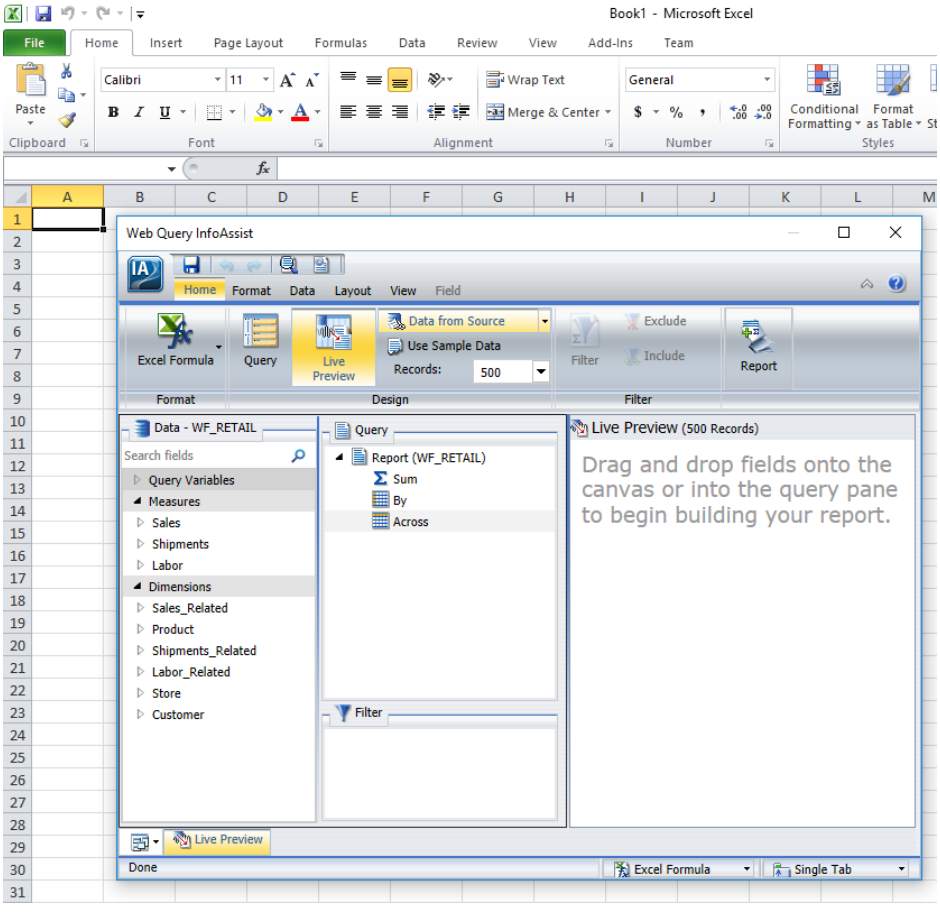
1. Open an Excel file, click the *Web Query* option in the Excel menu, then select *Create Query*.
2. Specify http://*hostname*:12331 in the Web Server URL field in the Web Server Connection dialog box that opens, then click *Next*.

   Note: Specify any Web Server URL that works in your reporting environment.
3. Select a folder to determine which Master Files (synonyms) may be used to create a query with InfoAssist+.
4. In the Data Source Selection dialog box that opens, select *CEN_ORDERS* from the Data Source list, then click *Finish*.

   InfoAssist+ opens.
5. In Interactive Design View, select *By* under Query on the left pane, and double-click *Product Category* from the Data list.

6. Select *Sum* under Query on the left pane, and double-click *Cost* and *Revenue* from the Data list.
7. On the Home tab, click the *Report* icon, and select *Column Totals*.

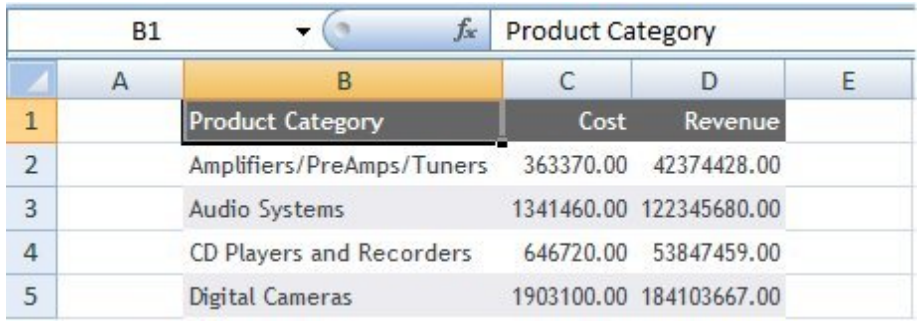
   After you perform the steps up to this point, the InfoAssist+ window looks similar to the following.



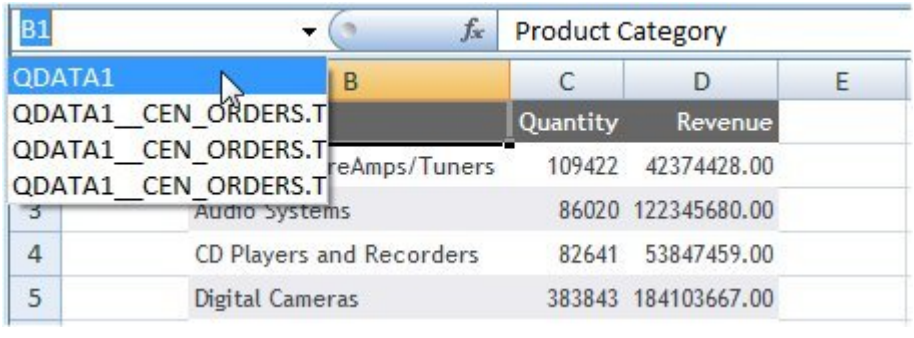
8. Click the *Save* button.

   If you selected the *Show Properties dialog when the query is created* option in the Db2 Web Query Spreadsheet Client dialog box, the Properties dialog box opens. It allows you to choose to insert the new data into the Excel worksheet, or to replace the existing data with the new data. For details, see *Installing and Setting Up the Spreadsheet Client* on page 47.
9. If you see the Properties dialog box, select *Insert cells for new data*, and click *OK*.

The report query data is returned to the Excel file, as shown in the following image.



10. Click the drop-down arrow to the right of the Name Box. You will see named ranges that are automatically added to the query, as shown in the following image.



Named ranges are added to the entire data table. The named range for the entire data table is QDATA1.

11. Select *QDATA1* from the Name Box. The data in the table is automatically highlighted.
12. Save the Excel file so that it can be reused in the example on editing an existing report query.

Editing Report Queries in InfoAssist+

You can edit an existing query previously created with Db2 Web Query Spreadsheet Client in an Excel file. The Edit Query option automatically launches InfoAssist+ using the same connection attributes and data source selected when the query was first created or last saved.

If you want to edit the connection attributes, prior to editing the query, right-click any cell in the existing query, and select *Edit Connection* to open the Web Server Connection dialog box.

Note:

- You cannot select a new Master File when editing an existing query.
- You cannot edit password-protected cells.

Procedure: How to Edit an Existing Report Query in InfoAssist+

1. Open the desired Excel file that contains the existing query.
2. Move the cursor over any cell in the existing query data, and select the *Edit Query* option from the Excel right-click context menu.

   If there is only one query in the Excel file, you can also use the Db2 Web Query menu in the main toolbar to select the *Edit Query* option.

   Selecting *Edit Query* launches InfoAssist+, where you can edit the existing query.

Any cell that contains data from the query is a part of that query, so that you can click anywhere within that range of data to edit the query.

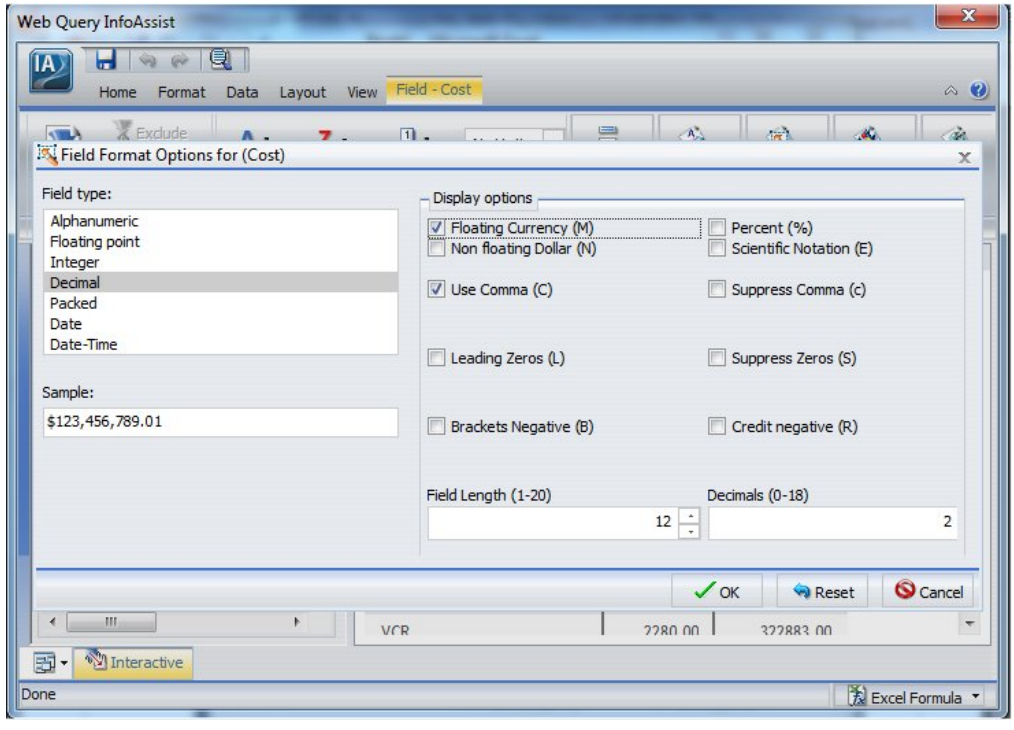
Example: Editing an Existing Report Query in InfoAssist+

This example covers multiple aspects of editing an existing report query using the Db2 Web Query Spreadsheet Client tool from an Excel file.

1. Open the existing Excel file created in the previous example in *How to Create a New Report Query in InfoAssist+* on page 61.
2. Right-click any cell in the data area of the existing report query, and select *Edit Query*.

   InfoAssist+ opens, displaying the query.
3. To change the format of the Cost measure, right-click *Cost* under Sum in the Query list, and click *Edit Format* to open the Field Format Options dialog box.
4. Set the Field type to *Decimal*, leave the Field Length set to the default value of *12*, set the number of Decimals to *2*, and select *Floating Currency (M)* and *Use Comma (C)* from the Display options list.

The Field Format Options dialog box will look similar to the following.



5. Click *OK*.
6. Perform the same three previous steps to change the format of the Revenue measure to match the settings applied to the Revenue measure.
7. Right-click *Product Category* under By in the Query list, and click *Delete* to delete this field.
8. Double-click *Product Type* in the Data list, making it a By field to replace the Product Category field that you deleted in the previous step.
9. Click the *Save* button.

The report query data is returned to the Excel file, as shown in the following image.

Tip: You may need to widen the Cost and Revenue columns to correctly display the TOTAL value.

10. Since you can do formatting in Db2 Web Query and Excel, you need to make sure that the Excel formatting is preserved when you edit a query and return the data to Excel.

    To preserve any formatting applied in Excel, right-click any cell in the data area of the query, and select *Data Range Properties*.

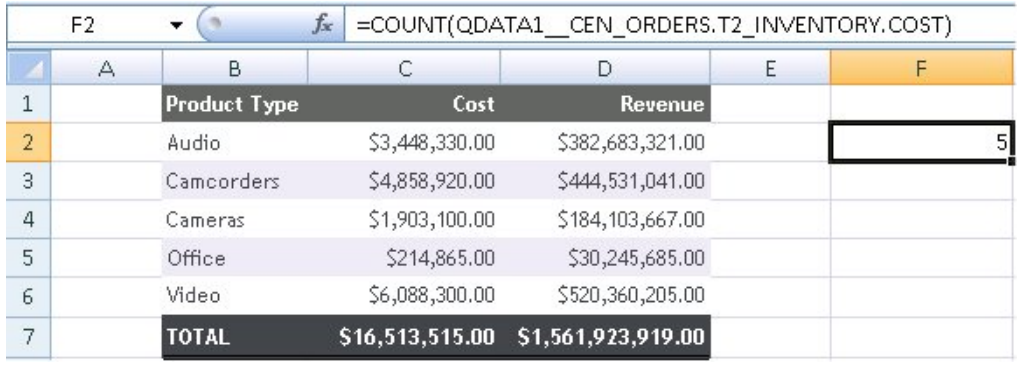
    The External Data Range Properties dialog box opens.

11. Select the *Preserve cell formatting* check box in the Data formatting and layout area of the dialog box, and click *OK*.

12. To demonstrate the value of having named ranges, add a formula that counts the number of values in a named range.

    Click any cell to the right of the existing query data, click the down arrow to the right of the AutoSum (formula) button on the Excel toolbar, select *Count Numbers*, and replace the range of cells in the function text box with the named range for the Cost column, which is QDATA1__CEN_ORDERS.T2_INVENTORY.COST. This cell now contains =COUNT(QDATA1__CEN_ORDERS.T2_INVENTORY.COST), and press the Enter key.

Because there are 5 values in the Cost column, the cell displays 5, as shown in the following image of the worksheet.



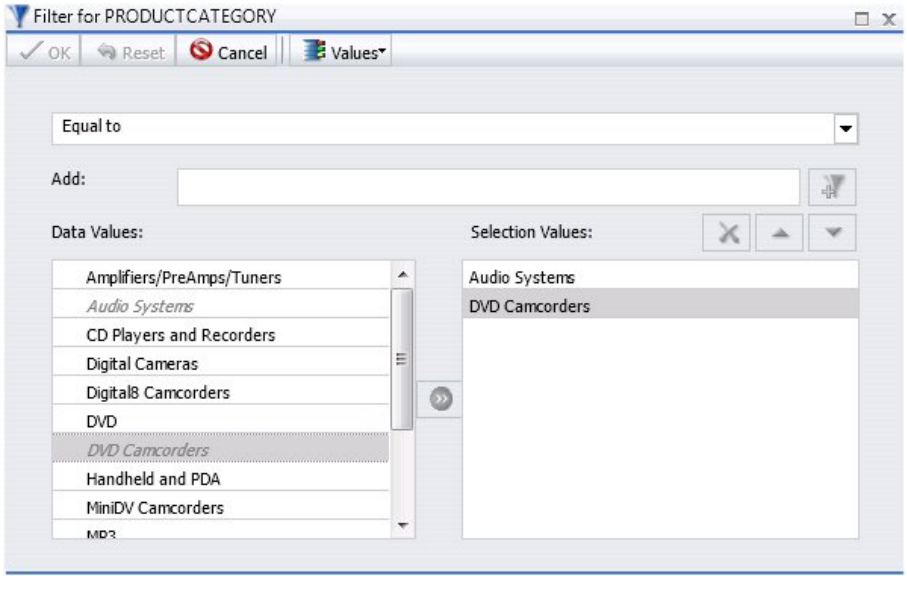
	A	B	C	D	E	F
1		Product Type	Cost	Revenue		
2		Audio	$3,448,330.00	$382,683,321.00		5
3		Camcorders	$4,858,920.00	$444,531,041.00		
4		Cameras	$1,903,100.00	$184,103,667.00		
5		Office	$214,865.00	$30,245,685.00		
6		Video	$6,088,300.00	$520,360,205.00		
7		TOTAL	$16,513,515.00	$1,561,923,919.00		

Notice the COUNT formula displayed in the function text box above the query data.

13. To edit the query again, right-click any cell in the query data, and select *Edit Query*. InfoAssist+ opens, displaying the query.

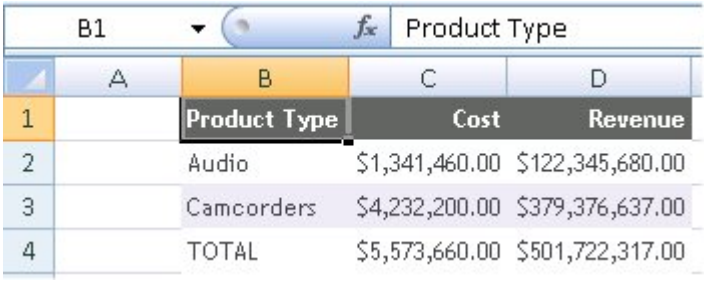
14. To add a filter, select *Product Category* from the Data list on the left pane, and click the *Filter* icon on the Home tab. In the Filter for PRODUCTCATEGORY dialog box, click the *Values* button, and select *Fetch All Values from Source*. Move *Audio Systems* and *DVD Camcorders* from the Data Values list to the Selection Values list, and click *OK*.

    The filter is shown in the following image.



15.Click the *Save* button.

The filtered data is returned to the Excel file, as shown in the following image.



16.Save the Excel file.

Creating Report Queries From Structured Ad hoc Forms

You can create a new report query directly from Excel by accessing existing SAFs (Structured Ad hoc Forms). A SAF is an HTML form containing a report procedure that is already connected to a data source, which enables you to select from a series of parameters to create output that is added to the active worksheet in Excel. You can analyze the output data in Excel and rerun the query to refresh the data as needed.

Note: SAFs are created in the Developer Workbench HTML Canvas, which references an existing parameterized report.

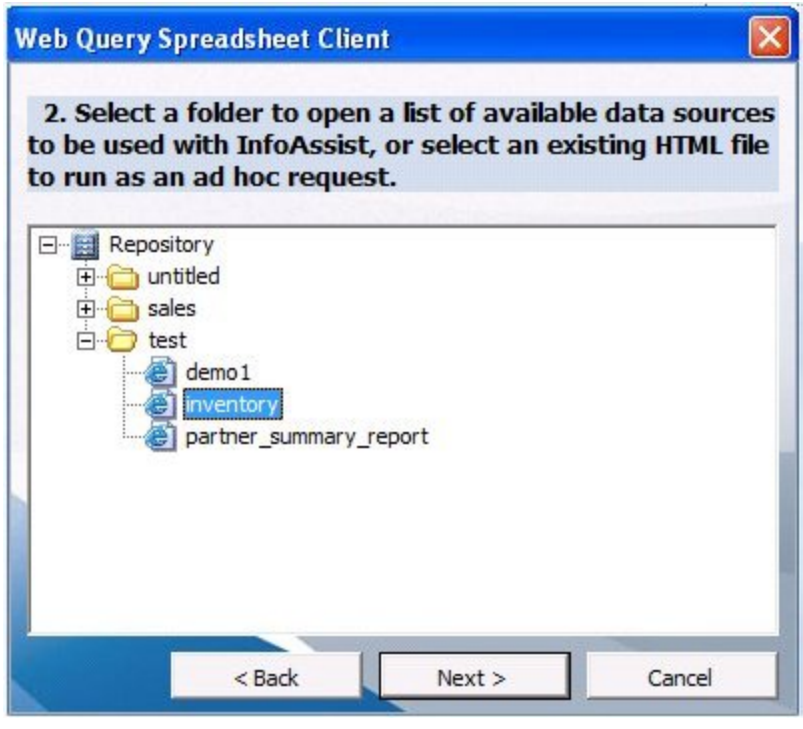
Procedure: How to Create a Report Query From a Structured Ad hoc Form

1. Open an Excel file.
2. Select a cell in which to place the query.
3. Click the *Web Query* option in the Excel menu, then select *Create Query*.

   You can also right-click any cell and select *Create Web Query Query*.
4. If connection information was not specified during the creation of a previous query, you must specify the desired Reporting Server URL when the Web Server Connection dialog box opens. When the desired connection settings have been specified, click *Next*. For more information, see *Defining Web Server Connection Settings* on page 57.

   Spreadsheet Client is configured to use Managed Reporting Authentication, so you are prompted to sign in with valid Db2 Web Query credentials (unless you are already signed in).

Structured ad hoc forms can now reside in any folder. Folders and HTML files will be listed based on user privileges, as shown in the following image.



5. Navigate to the folder, select an HTML file (SAF), and click *Next*.

6. Select a parameter in the Web Query HTML Form dialog box that appears, as shown in the following image. Next, click the *Run* button to populate the Excel file with data from the report.

The output data is added to the Excel file, as shown in the following image.

	A	B	C	D
1				
2	**Sales Analysis**			
3			Units	
4	City	Units Sold	Returned	Revenue
5	BOS	1275843	123134	$881,737,037.00
6	DAL	201193	19383	$141,089,667.00
7	LA	165755	28521	$108,025,795.00
8	ORL	213405	20445	$136,998,975.00
9	SEA	97242	9390	$69,131,048.00
10	STL	333143	31765	$224,941,397.00
11				
12	**TOTAL**	**2286581**	**232638**	**$1,561,923,919.00**

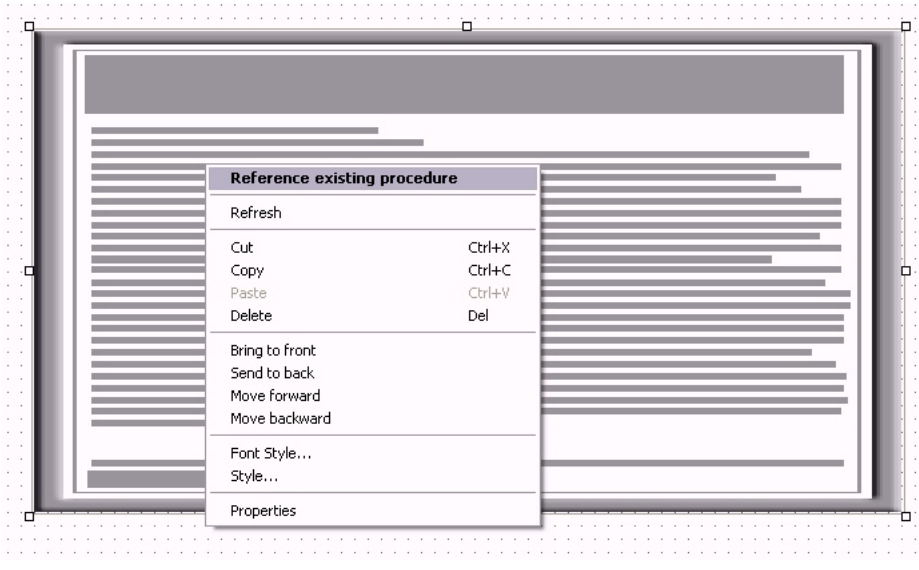
You can edit the query to select different parameter values, or just rerun the same parameter values to refresh the data, by right-clicking any cell in the query and selecting *Edit Query*.

Creating Structured Ad hoc Forms in Developer Workbench

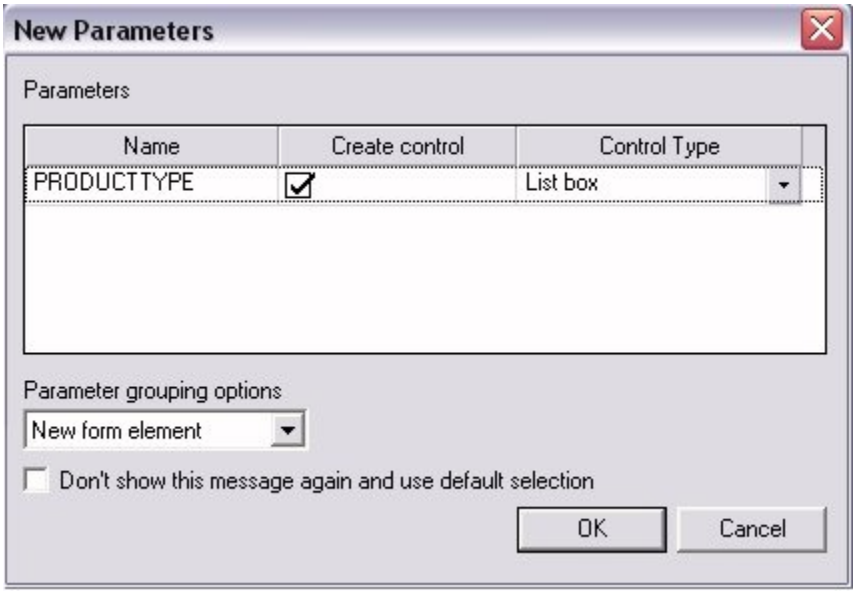
SAFs (Structured Ad hoc Forms) are created in the Developer Workbench Managed Reporting environment using the HTML Canvas, by referencing an existing parameterized Db2 Web Query report. A parameter must be added to a report to make it a valid SAF.

Procedure: How to Create a Structured Ad hoc Form

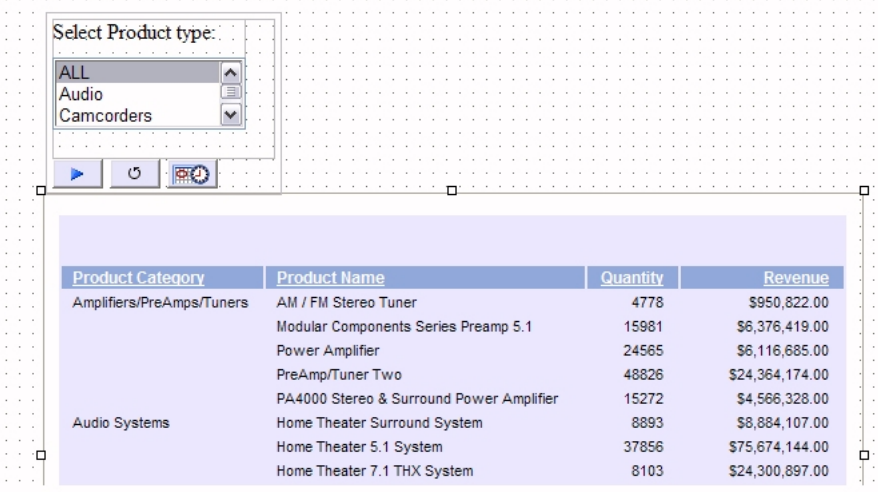
1. Insert a parameterized report into your HTML Layout by referencing an existing report, as shown in the following image.



2. Select the desired options in the New Parameters dialog box that appears.



The completed report that will be used as a SAF in Spreadsheet Client is shown in the following image.



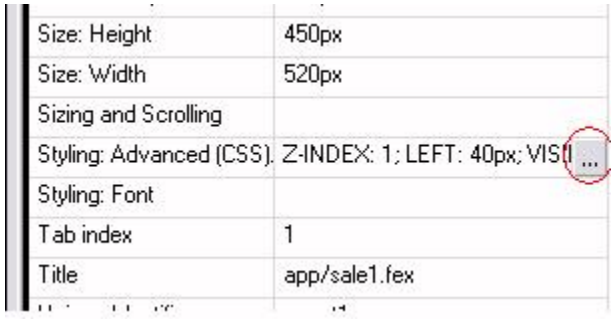
Product Category	Product Name	Quantity	Revenue
Amplifiers/PreAmps/Tuners	AM / FM Stereo Tuner	4778	$950,822.00
	Modular Components Series Preamp 5.1	15981	$6,376,419.00
	Power Amplifier	24565	$6,116,685.00
	PreAmp/Tuner Two	48826	$24,364,174.00
	PA4000 Stereo & Surround Power Amplifier	15272	$4,566,328.00
Audio Systems	Home Theater Surround System	8893	$8,884,107.00
	Home Theater 5.1 System	37856	$75,674,144.00
	Home Theater 7.1 THX System	8103	$24,300,897.00

When a SAF is called from Excel, Spreadsheet Client automatically bypasses the report frame in the HTML page and returns the data to Excel. This allows forms to be used in the browser and in Excel.

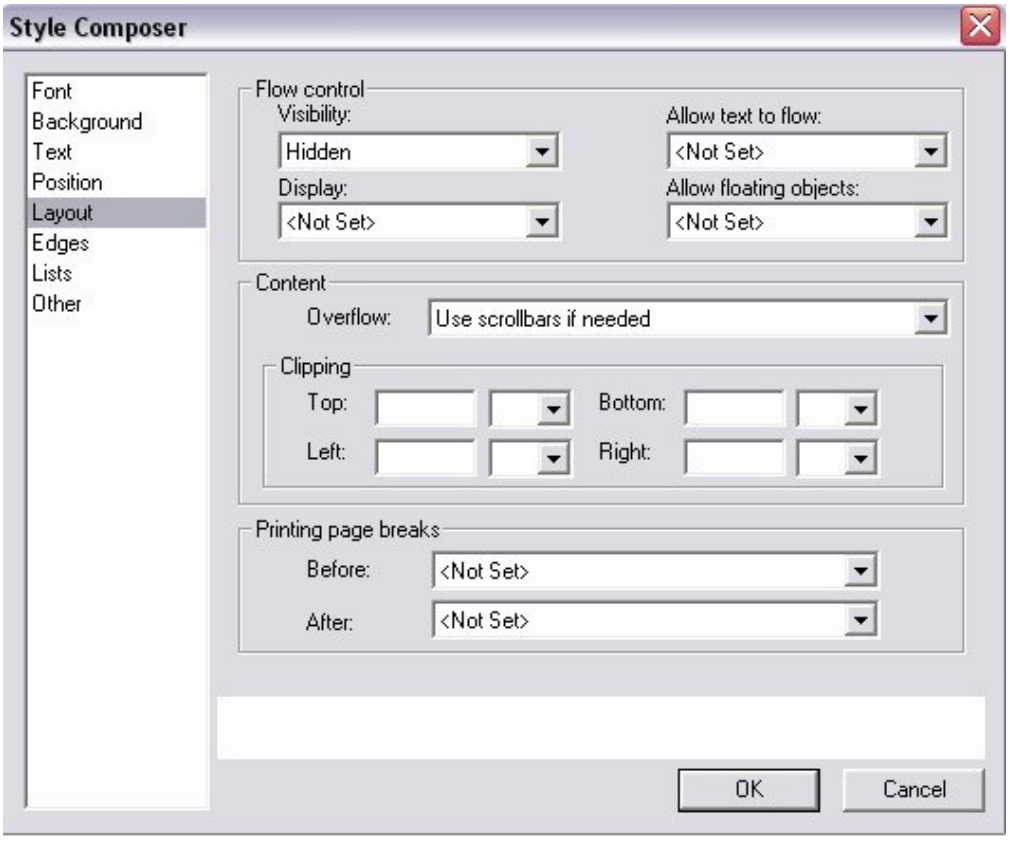
Procedure: How to Hide a Report Frame

If a form is to be used only in Excel, then you can hide the frame by changing the visibility property to hidden in the Layout window of the Style Composer in the HTML Canvas.

1. Highlight the report frame.
2. In the Properties panel, select the Styling Advance ellipsis button.

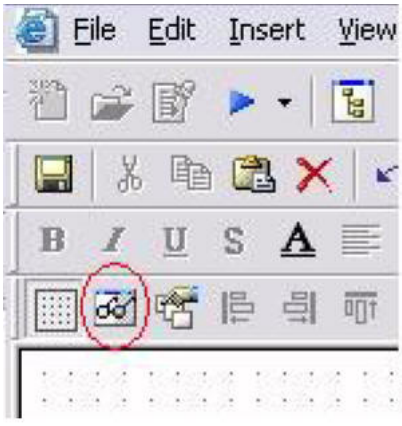


The Style Composer appears, as shown in the following image.



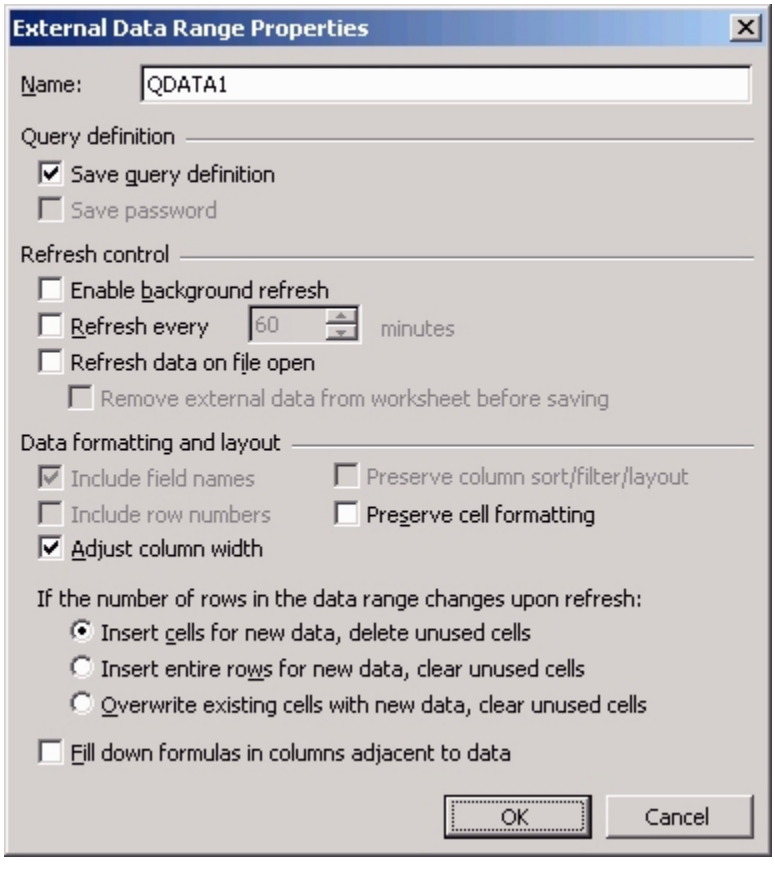
3. Select the *Layout* option.
4. Select *Hidden* in the Flow control Visibility drop-down list.
5. Click *OK*.

Once you hide a frame, you can make it visible in the HTML layout toolbar by clicking the Visibility Toggle button, as shown in the following image.



Setting Query Properties

You can modify query properties in Excel by right-clicking any cell in an existing query and selecting *Data Range Properties*, which opens the External Data Range Properties dialog box, as shown in the following image.



Some of the External Data Range Properties options you can control include:

- **Name.** You can change the name that is automatically assigned to the named range.

  Spreadsheet Client automatically adds a named range to the entire data table and also to each individual column. Named ranges are useful when referencing data as a source for analysis or within an advanced Excel application.

- **Query definition.** Unchecking this option will remove the query from the worksheet. The data remains but is no longer tied to a Spreadsheet Client query.

- **Refresh control.** You can enable background refresh, set the refresh interval, and enable the file to refresh data when opened. If applicable, you can also remove external data from the worksheet before saving.

Defining Data Formatting and Layout

The report layout determines how the data returned from the server interacts with the existing worksheet and any existing content within the worksheet. The following are options you can select in the Data formatting and layout section of the External Data Range Properties dialog box.

- **Preserve cell formatting.** This option affects how data is returned to the worksheet when refreshed. If this option is checked, the existing formatting in the worksheet is preserved. If this option is unchecked, the existing formatting is removed when the query is refreshed.
- **Auto adjustment of existing data.** These options determine how existing data is handled when new data is returned from the query. In some instances, the number of rows returned from the query is more or less than the original data set, and the following options determine what happens when this occurs.
  - Insert cells for new data, delete unused cells.
  - Insert entire rows for new data, clear unused cells.
  - Overwrite existing cells with new data, clear unused cells.

For more information on specifying external data range properties for a query, see your Microsoft Excel documentation.

Chapter 4

Describing an Individual Field

A field is the smallest meaningful element of data in a data source, but it can exhibit a number of complex characteristics. Master File attributes are used to describe these characteristics.

In this chapter:

Field Characteristics

The Master File describes the following field characteristics:

- The name of the field, as identified in the FIELDNAME attribute.
- Another name for the field, either its original name, as defined to its native data management system, or (for some types of data sources) a synonym of your own choosing, or (in some special cases) a pre-defined value that tells how to interpret the field, that you can use as an alternative name in requests. This alternative name is defined by the ALIAS attribute.

- How the field stores and displays data, specified by the ACTUAL, USAGE, and MISSING attributes.

  The ACTUAL attribute describes the type and length of the data as it is actually stored in the data source. For example, a field might be alphanumeric and 15 characters in length. Note that FOCUS data sources do not use the ACTUAL attribute, and instead use the USAGE attribute to describe the data as it is formatted. handles the storage.

  The USAGE attribute, also known by its alias, FORMAT, describes how a field is formatted when it appears in reports. You can also specify edit options, such as date formats, floating dollar signs, and zero suppression.

  The MISSING attribute enables null values to be entered into and read from a field in data sources that support null data, such as FOCUS data sources and most relational data sources.

- The option for a field to be virtual, rather than being stored in the data source, and have its value derived from information already in the data source. Virtual fields are specified by the DEFINE attribute.
- Optional field documentation for the developer, contained in the DESCRIPTION attribute.
- Acceptable data-entry values for the field, specified by the ACCEPT attribute.
- An alternative report column title for the field, described by the TITLE attribute.
- A 100-year window that assigns a century value to a two-digit year stored in the field. Two attributes define this window: DEFCENT and YRTHRESH.

The Field Name: FIELDNAME

Identify a field using FIELDNAME, the first attribute specified in a field declaration in the Master File. You can assign any name to a field, regardless of its name in its native data source. Likewise, for FOCUS data sources, you can assign any name to a field in a new data source.

When you generate a report, each column title in the report has the name of the field displayed in that column as its default, so assigning meaningful field names helps readers of the report. Alternatively, you can specify a different column title within a given report by using the AS phrase in the report request, as described in the manual, or a different default column title for all reports by using the TITLE attribute in the Master File, as described in *Alternative Report Column Titles: TITLE* on page 145.

Syntax: How to Identify the Field Name

```
FIELD[NAME] = field_name
```

where:

field_name
Is the name you are giving this field. It can be a maximum of 512 characters. Some restrictions apply to names longer than 12 characters, as described in *Restrictions for Field Names* on page 82. The name can include any combination of letters, digits, and underscores (_), and must contain at least one letter. Other characters are not recommended, and may cause problems in some operating environments or when resolving expressions.

It is recommended that you not use field names of the type Cn, En, and Xn (where n is any sequence of one or two digits) because these can be used to refer to report columns, HOLD file fields, and other special objects.

If you must use special characters because of a field report column title, consider using the TITLE attribute in the Master File to specify the title, as described in *Alternative Report Column Titles: TITLE* on page 145.

Reference: Usage Notes for FIELDNAME

Note the following rules when using FIELDNAME:

- **Alias.** FIELDNAME has an alias of FIELD.
- **Changes.** In a FOCUS data source, if the INDEX attribute has been set to I (that is, if an index has been created for the field), you cannot change the field name without rebuilding the data source. You may change the name in all other situations.

Reference: Restrictions for Field Names

The following restrictions apply to field names and aliases longer than 12 characters:

- You cannot use a field name longer than 12 characters to specify a cross-referenced field in a JOIN command when the cross-referenced file is a FOCUS data source.
- Indexed fields and text fields in FOCUS data sources cannot have field names longer than 12 characters. Indexed fields and text fields in XFOCUS data sources are not subject to this 12 character limitation. Long ALIAS names are supported for both types of data sources.
- A field name specified in an alternate file view cannot be qualified.
- The CHECK FILE command PICTURE and HOLD options display the first 11 characters of long names within the resulting diagram or HOLD file. A caret (>) in the 12th position indicates that the name is longer than the displayed portion.
- ?FF, ? HOLD, ? DEFINE

  These display up to 31 characters of the name, and display a caret (>) in the 32nd character to indicate a longer field name.

Using a Qualified Field Name

Requests can qualify all referenced field names and aliases with file and/or segment names, a useful technique when duplicate field names exist across segments in a Master File or in joined data sources.

The names of text fields and indexed fields in FOCUS Master Files are limited to 12 characters. Text fields and indexed fields in XFOCUS Master Files are not subject to this 12-character limitation. However, the aliases for text and indexed fields may be up to 512 characters. Field names up to 512 characters appear as column titles in TABLE reports if there is no TITLE attribute or AS phrase.

The default value for the SET FIELDNAME command, SET FIELDNAME=NEW, activates long and qualified field names. The syntax is described in the manual.

Syntax: How to Specify a Qualified Field Name in a Request

```
[filename.][segname.]fieldname
```

where:

filename

Is the name of the Master File or tag name. Tag names are used with the JOIN and COMBINE commands.

segname

Is the name of the segment in which the field resides.

fieldname

Is the name of the field.

Example: **Qualifying a Field Name**

The fully qualified name of the field EMP_ID in the EMPINFO segment of the EMPLOYEE data source is:

```
EMPLOYEE.EMPINFO.EMP_ID
```

Syntax: **How to Change the Qualifying Character**

```
SET QUALCHAR = qualcharacter
```

The period (.) is the default qualifying character.

Using a Duplicate Field Name

Field names are considered duplicates when you can reference two or more fields with the same field name or alias. Duplication may occur:

- If a name appears multiple times within a Master File.
- In a JOIN between two or more Master Files, or in a recursive JOIN.
- If you issue a COMBINE and do not specify a prefix.

Duplicate fields (those having the same field name and alias) are not allowed in the same segment. The second occurrence is never accessed, and the following message is generated when you issue CHECK and CREATE FILE:

```
(FOC1829) WARNING. FIELDNAME IS NOT UNIQUE WITHIN A SEGMENT: fieldname
```

Duplicate field names may exist across segments in a Master File. To retrieve such a field, you must qualify its name with the segment name in a request. If a field that appears multiple times in a Master File is not qualified in a request, the first field encountered in the Master File is the one retrieved.

Note: If a Master File includes duplicate field names for real fields and/or virtual fields, the following logic is used when retrieving a field:

- If only virtual fields are duplicated, the last virtual field is retrieved.
- If only real fields are duplicated, the first real field is retrieved.
- If a Master File has both a real field and one or more virtual fields with the same name, the last virtual field is retrieved.
- If a field defined outside of a Master File has the same name as a virtual or real field in a Master File, the last field defined outside of the Master File is retrieved.

Reports can include qualified names as column titles. The SET QUALTITLES command, discussed in the manual, determines whether reports display qualified column titles for duplicate field names. With SET QUALTITLES=ON, they display qualified column titles for duplicate field names even when the request itself does not specify qualified names. The default value, OFF, disables qualified column titles.

Rules for Evaluating a Qualified Field Name

The following rules are used to evaluate qualified field names:

- The maximum field name qualification is *filename.segname.fieldname*. For example:

```
TABLE FILE EMPLOYEE
PRINT EMPLOYEE.EMPINFO.EMP_ID
END
```

includes EMP_ID as a fully qualified field. The file name, EMPLOYEE, and the segment name, EMPINFO, are the field qualifiers.

Qualifier names can also be duplicated. For example:

```
FILENAME=CAR, SUFFIX=FOC
  SEGNAME=ORIGIN, SEGTYPE=S1
      FIELDNAME=COUNTRY, COUNTRY, A10, $
  SEGNAME=COMP, SEGTYPE=S1, PARENT=ORIGIN
      FIELDNAME=CAR, CARS, A16, $
        .
        .
        .
TABLE FILE CAR
PRINT CAR.COMP.CAR
END
```

This request prints the field with alias CARS. Both the file name and field name are CAR.

A field name can be qualified with a single qualifier, either its file name or its segment name. For example:

```
FILENAME=CAR, SUFFIX=FOC
   SEGNAME=ORIGIN, SEGTYPE=S1
      FIELDNAME=COUNTRY, COUNTRY, A10, $
   SEGNAME=COMP, SEGTYPE=S1, PARENT=ORIGIN
      FIELDNAME=CAR, CARS, A16, $
        .
        .
        .
TABLE FILE CAR
PRINT COMP.CAR AND CAR.CAR
END
```

This request prints the field with alias CARS twice.

When there is a single qualifier, segment name takes precedence over file name. Therefore, if the file name and segment name are the same, the field qualified by the segment name is retrieved.

- If a field name begins with characters that are the same as the name of a prefix operator, it may be unclear whether a request is referencing that field name or a second field name prefixed with the operator. The value of the first field is retrieved, not the value calculated by applying the prefix operator to the second field. In the next example, there is a field whose unqualified field name is CNT.COUNTRY and another whose field name is COUNTRY:

```
FILENAME=CAR, SUFFIX=FOC
   SEGNAME=ORIGIN, SEGTYPE=S1
      FIELDNAME=CNT.COUNTRY, ACNTRY, A10, $
      FIELDNAME=COUNTRY, BCNTRY, A10, $
TABLE FILE CAR
SUM CNT.COUNTRY
END
```

  In this request, the string CNT.COUNTRY is interpreted as a reference to the field named CNT.COUNTRY, not as a reference to the prefix operator CNT. applied to the field named COUNTRY. Therefore, the request sums the field whose alias is ACNTRY. Although the field name CNT.COUNTRY contains a period as one of its characters, it is an unqualified field name. It is not a qualified name or a prefix operator acting on a field name, neither of which is allowed in a Master File. The request does not count instances of the field whose alias is BCNTRY.

- If a Master File has either a file name or segment name that is the same as a prefix operator, the value of the field within the segment is retrieved in requests, not the value calculated by applying the prefix operator to the field.

  For example:

```
FILENAME=CAR, SUFFIX=FOC
   SEGNAME=ORIGIN, SEGTYPE=S1
      FIELDNAME=COUNTRY, COUNTRY, A10, $
   SEGNAME=PCT, SEGTYPE=S1, PARENT=ORIGIN
      FIELDNAME=CAR, CARS, I2, $
 TABLE FILE CAR
SUM PCT.CAR PCT.PCT.CAR
BY COUNTRY
END
```

  This request sums the field with alias CARS first, and then the percent of CARS by COUNTRY.

- When a qualified field name can be evaluated as a choice between two levels of qualification, the field name with the higher level of qualification takes precedence.

  In the following example, the choice is between an unqualified field name (the field named ORIGIN.COUNTRY in the ORIGIN segment) and a field name with segment name qualification (the field named COUNTRY in the ORIGIN segment). The field with segment name qualification is retrieved:

```
FILENAME=CAR, SUFFIX=FOC
   SEGNAME=ORIGIN, SEGTYPE=S1
      FIELDNAME=ORIGIN.COUNTRY, OCNTRY, A10, $
      FIELDNAME=COUNTRY, CNTRY, A10, $
TABLE FILE CAR
PRINT ORIGIN.COUNTRY
END
```

  This request prints the field with alias CNTRY. To retrieve the field with alias OCNTRY, qualify its field name, ORIGIN.COUNTRY, with its segment name, ORIGIN:

```
PRINT ORIGIN.ORIGIN.COUNTRY
```

- When a qualified field name can be evaluated as a choice between two field names with the same level of qualification, the field with the shortest basic field name length is retrieved. For example:

```
FILENAME=CAR, SUFFIX=FOC
   SEGNAME=CAR, SEGTYPE=S1
      FIELDNAME=CAR.CAR, CAR1, A10, $
   SEGNAME=CAR.CAR, SEGTYPE=S1, PARENT=CAR
      FIELDNAME=CAR, CAR2, A10, $
TABLE FILE CAR
PRINT CAR.CAR.CAR
END
```

  In this example, it is unclear if you intend CAR.CAR.CAR to refer to the field named CAR.CAR in the CAR segment, or the field named CAR in the CAR.CAR segment. (In either case, the name CAR.CAR is an unqualified name that contains a period, not a qualified name. Qualified names are not permitted in Master Files.)

  No matter what the intention, the qualified field name is exactly the same and there is no obvious choice between levels of qualification.

  Since the field with alias CAR2 has the shortest basic field name length, CAR2 is printed. This is different from the prior example, where the choice is between two levels of qualification. To retrieve the CAR1 field, you must specify its alias.

The Field Synonym: ALIAS

You can assign every field an alternative name, or alias. A field alias may be its original name as defined to its native data source, any name you choose, or, in special cases, a predefined value. The way in which you assign the alias is determined by the type of data source and, in special cases, the role the field plays in the data source. After it has been assigned, you can use this alias in requests as a synonym for the regular field name. Assign this alternative name using the ALIAS attribute.

Example: Using a Field Synonym

In the EMPLOYEE data source, the name CURR_SAL is assigned to a field using the FIELDNAME attribute, and the alternative name CSAL is assigned to the same field using the ALIAS attribute:

```
FIELDNAME = CURR_SAL, ALIAS = CSAL, USAGE = D12.2M,  $
```

Both names are equally valid within a request. The following TABLE requests illustrate that they are functionally identical, refer to the same field, and produce the same result:

```
TABLE FILE EMPLOYEE
PRINT CURR_SAL BY EMP_ID
END

TABLE FILE EMPLOYEE
PRINT CSAL BY EMP_ID
END
```

Note: In extract files (HOLD, PCHOLD), the field name is used to identify fields, not the ALIAS.

Implementing a Field Synonym

The value you assign to ALIAS must conform to the same naming conventions to which the FIELDNAME attribute is subject, unless stated otherwise. Assign a value to ALIAS in the following way for the following types of data sources:

- **Relational data sources.** ALIAS describes the field original column name as defined in the relational table.
- **Sequential data sources.** ALIAS describes a synonym, or alternative name, that you can use in a request to identify the field. You can assign any name as the alias. Many users choose a shorter version of the primary name of the field. For example, if the field name is LAST_NAME, the alias might be LN. The ALIAS attribute is required in the Master File, but it can have the value blank.

Note that ALIAS is used in a different way for sequenced repeating fields, where its value is ORDER, as well as for RECTYPE and MAPVALUE fields when the data source includes multiple record types.

- **FOCUS data sources.** ALIAS describes a synonym, or alternative name, that you can use in a request to identify the field. You can assign any name as the alias. Many users choose a shorter version of the primary name of the field. For example, if the field name is LAST_NAME, the alias might be LN. The ALIAS attribute is required in the Master File, but it can have the value blank. Aliases can be changed without rebuilding the data source. If an alias is referred to in other data sources, similar changes may be needed in those Master Files.

The Displayed Data Type: USAGE

This attribute, which is also known as FORMAT, describes how to format a field when displaying it in a report or using it in a calculation.

Specifying a Display Format

For FOCUS data sources, which do not use the ACTUAL attribute, USAGE also specifies how to store the field. For other types of data sources, assign a USAGE value that corresponds to the ACTUAL value, to identify the field as the same data type used to store it in the data source. If the data is store as alphanumeric, assign the USAGE based on how the field will be displayed in your reports. The conversion is done automatically. For instructions on which ACTUAL values correspond to which USAGE values, see the documentation for the specific data adapter.

In addition to selecting the data type and length, you can also specify display options, such as date formatting, floating dollar signs, and zero suppression. Use these options to customize how the field appears in reports.

Syntax: How to Specify a Display Format

```
USAGE = tl[d]
```

where:

t

Is the data type. Valid values are A (alphanumeric), F (floating-point single-precision), D (floating-point double-precision), I (integer), P (packed decimal), D, W, M, Q, or Y used in a valid combination (date), and TX (text).

l

Is a length specification. The specification varies according to the data type. See the section for each data type for more information. Note that you do not specify a length for date format fields.

d

Is one or more display options. Different data types offer different display options. See the section for each data type for more information.

The complete USAGE value cannot exceed eight characters.

The values that you specify for type and field length determine the number of print positions allocated for displaying or storing the field. Display options only affect displayed or printed fields. They are not active for non-display retrievals, such as extract files.

Note: If a numeric field cannot display with the USAGE format given (for example, the result of aggregation is too large), asterisks appear.

See the sections for each format type for examples and additional information.

Reference: Usage Notes for USAGE

Note the following rules when using USAGE:

- **Alias.** USAGE has an alias of FORMAT.
- **Changes.** For most data sources, you can change the type and length specifications of USAGE only to other types and lengths valid for the ACTUAL attribute of that field. You can change display options at any time.

  For FOCUS data sources, you cannot change the type specification. You can change the length specification for I, F, D, and P fields, because this affects only display, not storage. You cannot change the decimal part of the length specification for P fields. You can change the length specification of A (alphanumeric) fields only if you use the REBUILD facility. You can change display options at any time.

Data Type Formats

You can specify several types of formats:

- **Numeric.** There are four types of numeric formats: integer, floating-point single-precision, floating-point double-precision, and packed decimal. See *Numeric Display Options* on page 95 for additional information.

- **Alphanumeric.** You can use alphanumeric format for any value to be interpreted as a sequence of characters and composed of any combination of digits, letters, and other characters.
- **Date.** The date format enables you to define date components, such as year, quarter, month, day, and day of week to:
  - Sort by date.
  - Do date comparisons and arithmetic with dates.
  - Validate dates automatically in transactions.

  Note that for some applications, such as assigning a date value using the DECODE function, you may wish instead to use alphanumeric, integer, or packed-decimal fields with date display options, which provide partial date functionality.
- **Date-Time.** The date-time format supports both the date and the time, similar to the timestamp data types available in many relational data sources. Date-time fields are stored in eight, ten, or 12 bytes: four bytes for date and either four, six, or eight bytes for time, depending on whether the format specifies a microsecond or nanosecond. Computations only allow direct assignment within data types. All other operations are accomplished through a set of date-time functions.
- **Text.** Text fields can be used to store large amounts of data and display it with line breaks.

Integer Format

You can use integer format for whole numbers. An integer is any value composed of the digits zero to nine, without a decimal point.

You can also use integer fields with date display options to provide limited date support. This use of integer fields is described in the *Alphanumeric and Numeric Formats With Date Display Options* on page 118.

The integer USAGE type is I. See *Numeric Display Options* on page 95. The format of the length specification is:

n

where:

n

Is the number of digits to display. The maximum length is 11, which must include the digits and a leading minus sign if the field contains a negative value. You can also specify a number of decimal places (up to *n* - 1), and the number will display with a decimal point before that number of digits.

For example:

Format	Display
I6	4316
I6.2	43.16
I2	22
I4	-617

Floating-Point Double-Precision Format

You can use floating-point double-precision format for any value composed of the digits zero to nine and an optional decimal point.

The floating-point double-precision USAGE type is D. See *Numeric Display Options* on page 95 for the compatible display options. The length specification format is:

t[.*s*]

where:

t

Is the number of characters to display up to a maximum of 33, including the numeric digits, an optional decimal point, and a leading minus sign if the field contains a negative value. The number of significant digits supported varies with the operating environment.

s

Is the number of digits that follow the decimal point. It can be a maximum of 31 and must be less than *t*.

For example:

Format	Display
D8.2	3,187.54
D8	416

In the case of D8.2, the 8 represents the maximum number of places, including the decimal point and decimal places. The 2 represents how many of these eight places are decimal places. The commas are automatically included in the display, and are not counted in the total.

Floating-Point Single-Precision Format

You can use floating-point single-precision format for any number, including numbers with decimal positions. The number is composed of the digits 0 to 9, including an optional decimal point. This format is intended for use with smaller decimal numbers. Unlike floating-point double-precision format, its length cannot exceed nine positions.

The floating-point single-precision USAGE type is F. Compatible display options are described in *Numeric Display Options* on page 95. The length specification format is:

t[.*s*]

where:

t

Is the number of characters to display, up to a maximum of 33, including the numeric digits, an optional decimal point, and a leading minus sign if the field contains a negative value. The number of significant digits supported varies with the operating environment.

s

Is the number of digits that follow the decimal point. It can be up to 31 digits and must be less than *t*.

For example:

Format	Display
F5.1	614.2

Format	Display
F4	318

Packed-Decimal Format

You can use packed-decimal format for any number, including decimal numbers. A decimal number is any value composed of the digits zero to nine, including an optional decimal point.

You can also use packed-decimal fields with date display options to provide limited date support. See *Alphanumeric and Numeric Formats With Date Display Options* on page 118.

The packed-decimal USAGE type is P. The compatible display options are described in *Numeric Display Options* on page 95.

The length specification format is:

t[.*s*]

where:

t

Is the number of characters to display, up to 33, including a maximum of 31 digits, an optional decimal point, and a leading minus sign if the field contains a negative value.

s

Is the number of digits that follow the decimal point. It can be up to 31 digits and must be less than *t*.

For example:

Format	Display
P9.3	4168.368
P7	617542

P fields have two internal lengths, 8 bytes (which supports up to 15 digits) and 16 bytes (which supports up to 33 digits). A USAGE of P1 through P15 is automatically assigned an internal storage consisting of 8 bytes. A USAGE of P16 or greater is assigned an internal storage consisting of 16 bytes.

If your USAGE does not account for the number of digits required to display the stored number, asterisks display instead of a number. This does not necessarily indicate an overflow of the field, just that you did not account for displaying the number of digits that are stored in the field.

Overflow occurs if you attempt to store a number with more digits than can actually fit in the internal storage assigned. Overflow such as this is indicated by storing a number consisting of all 9's, in all operating environments except z/OS. On z/OS, the value 0 (zero) is used. Therefore, if you try to store a number consisting of 16 digits in a packed field assigned 8 bytes of internal storage, the number 999999999999999 (the digit 9 repeated 15 times), or the number zero on z/OS, will be stored in the field instead.

If you assign a USAGE of P1 through P14 to such a field, the 15 digits stored in the field will not be able to be displayed, and you will see asterisks. However, if you assign the USAGE P15 to the field, it will be able to display the 15-digit number stored in the field, so you will see the value 999999999999999 (zero on z/OS). If you see that number for a P15 field, it could be the actual number that was required or it could be a replacement for a number that could not fit.

Numeric Display Options

Display options may be used to edit numeric formats. These options only affect how the data in the field is printed or appears on the screen, not how it is stored in your data source.

Edit Option	Meaning	Effect
-	Minus sign	Displays a minus sign to the right of negative numeric data. **Note:** Not supported with format options B, E, R, T, DMY, MDY, and YMD.
p	Percentage	Converts a number to a percentage by multiplying it by 100, and displays it followed by a percent sign (%). **Note:** This option is not supported for output formats XLSX and EXL2K.
%	Percent sign	Displays a percent sign (%), along with numeric data. Does not calculate the percent.

Edit Option	Meaning	Effect
A	Negative suppression	Displays the absolute value of the number, but does not affect the stored value. ❑ If you propagate a field with a negative suppression USAGE attribute to a HOLD file, the HOLD file contains the signed values. The negative suppression USAGE attribute is also propagated to the HOLD file so that if you run a report request against the HOLD file, the minus signs are suppressed on the report output. ❑ The negative suppression option cannot be used in with the following display options: ❑ B (bracket negative). ❑ R (credit negative). ❑ - (right side negative).
B	Bracket negative	Encloses negative numbers in parentheses.
C	Comma suppress	Suppresses the display of commas. Used with numeric format options M and N (floating and non-floating dollar sign) and data format D (floating-point double-precision).
C	Comma edit	Inserts a comma after every third significant digit, or a period instead of a comma if continental decimal notation is in use.
DMY	Day-Month-Year	Displays alphanumeric or integer data as a date in the form day/month/year.
E	Scientific notation	Displays only significant digits.
L	Leading zeroes	Adds leading zeroes.

Edit Option	Meaning	Effect
M	Floating currency symbol ($ for US code page)	Places a floating currency symbol to the left of the highest significant digit. The default currency symbol depends on the code page. You can use the SET CURRSYMB=*symbol* command to specify up to four characters as the currency symbol or one of the following currency codes: USD or '$' specifies U. S. dollars. GBP specifies the British pound. JPY specifies the Japanese yen or Chinese yuan. EUR specifies the Euro.
MDY	Month-Day-Year	Displays alphanumeric or integer data as a date in the form month/day/year.
N	Fixed currency symbol ($ for US code page)	Places a currency symbol to the left of the field. The symbol appears only on the first detail line of each page. The default currency symbol depends on the code page. You can use the SET CURRSYMB=*symbol* command to specify up to four characters as the currency symbol or one of the following currency codes: USD or '$' specifies U. S. dollars. GBP specifies the British pound. JPY specifies the Japanese yen or Chinese yuan. EUR specifies the Euro.
R	Credit (CR) negative	Places CR after negative numbers.
S	Zero suppress	If the data value is zero, prints a blank in its place.
T	Month translation	Displays the month as a three-character abbreviation.

Edit Option	Meaning	Effect
YMD	Year-Month-Day	Displays alphanumeric or integer data as a date in the form year/month/day.

Example: **Using Numeric Display Options**

The following table shows examples of the display options that are available for numeric fields.

Option	Format	Data	Display
Minus sign	I2- D7- F7.2-	-21 -6148 -8878	21- 6148- 8878.00-
Percent sign	I2% D7% F3.2%	21 6148 48	21% 6,148% 48.00%
Comma suppression	D6c D7Mc D7Nc	41376 6148 6148	41376 $6148 $ 6148
Comma inclusion	I6C	41376	41,376
Zero suppression	D6S	0	
Bracket negative	I6B	-64187	(64187)
Credit negative	I8R	-3167	3167 CR
Leading zeroes	F4L	31	0031
Floating dollar	D7M	6148	$6,148
Non-floating dollar	D7N	5432	$ 5,432
Scientific notation	D12.5E	1234.5	0.12345D+04
Year/month/day	I6YMD I8YYMD	980421 19980421	98/04/21 1998/04/21

Option	Format	Data	Display
Month/day/year	I6MDY I8MDYY	042198 04211998	04/21/98 04/21/1998
Day/month/year	I6DMY I8DMYY	210498 21041998	21/04/98 21/04/1998
Month translation	I2MT	07	JUL

Several display options can be combined, as shown:

Format	Data	Display
I5CB	-61874	(61,874)

All of the options may be specified in any order. Options M and N (floating and non-floating dollar sign) and data format D (floating-point double-precision) automatically invoke option C (comma). Options L and S cannot be used together. Option T (Translate) can be included anywhere in an alphanumeric or integer USAGE specification that includes the M (month) display option. Date display options (D, M, T, and Y), which cannot be used with floating-point fields, are described in *Alphanumeric and Numeric Formats With Date Display Options* on page 118.

Extended Currency Symbol Display Options

You can select a currency symbol for display in report output regardless of the default currency symbol configured for National Language Support (NLS). Use the extended currency symbol format in place of the floating dollar (M) or non-floating dollar (N) display option. When you use the floating dollar (M) or non-floating dollar (N) display option, the currency symbol associated with the default code page is displayed. For example, when you use an American English code page, the dollar sign is displayed.

The extended currency symbol format allows you to display a symbol other than the dollar sign. For example, you can display the symbol for a United States dollar, a British pound, a Japanese yen or Chinese yuan, or the euro. Extended currency symbol support is available for numeric formats (I, D, F, and P).

The extended currency symbol formats are specified as two-character combinations *in the last positions* of any numeric display format. The first character in the combination can be either an exclamation point (!) or a colon (:). The colon is the recommended character because it will work in all ASCII and EBCDIC code pages. The exclamation point is not consistent on all EBCDIC code pages and may produce unexpected behavior if the code page you are using translates the exclamation point differently.

In addition, you can use the SET commands SET CURSYM_D, SET CURSYM_E, SET CURSYM_F, SET CURSYM_G, SET CURSYM_L, and SET CURSYM_Y to redefine the default display characters for the extended currency symbol formats. For example, you can display a euro symbol on the right of the number and add a space between the number and the euro symbol by issuing the SET CURSYM_F command and using the extended currency symbol format :F in the request or Master File.

```
SET CURSYM_F = ' €'
```

For more information, see the *Developing Applications* manual.

The following table lists the supported extended currency display options:

Display Option	Description	Example
:d or !d	Fixed dollar sign.	D12.2:d
:D or !D	Floating dollar sign.	D12.2:D
:e or !e	Fixed euro symbol.	F9.2:e
:E or !E	Floating euro symbol on the left side.	F9.2:E
:F or !F	Floating euro symbol on the right side.	F9.2:F
:G or !G	Floating dollar symbol on the right side.	F9.2:G
:l or !l	Fixed British pound sign.	D12.1:l
:L or !L	Floating British pound sign.	D12.1:L
:y or !y	Fixed Japanese yen or Chinese yuan symbol.	I9:y

Display Option	Description	Example
:Y or !Y	Floating Japanese yen or Chinese yuan symbol.	I9:Y

Reference: Extended Currency Symbol Formats

The following guidelines apply:

- A format specification cannot be longer than eight characters.
- The extended currency option must be the last option in the format.
- The extended currency symbol format cannot include the floating (M) or non-floating (N) display option.
- A non-floating currency symbol is displayed only on the first row of a report page. If you use field-based reformatting (as in the example that follows) to display multiple currency symbols in a report column, only the symbol associated with the first row is displayed. In this case, do not use non-floating currency symbols.
- Lowercase letters are transmitted as uppercase letters by the terminal I/O procedures. Therefore, the fixed extended currency symbols can only be specified in a procedure.
- Extended currency symbol formats can be used with fields in floating point, decimal, packed, and integer formats. Alphanumeric and variable character formats cannot be used.

Reference: Locale Display Options for Currency Values

The CURRENCY_ISO_CODE and CURRENCY_DISPLAY 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.

Note: These parameters are not supported with FORMAT EXL2K. They are supported with FORMAT XLSX, FORMAT XLSX FORMULA, and FORMAT XLSX TEMPLATE.

The syntax is:

```
fld/fmt:C(CURRENCY_DISPLAY='pos',
   CURRENCY_ISO_CODE='iso')= expression;
```

where:

fld
Is the field to which the parameters are to be applied.

fmt
Is a numeric format that supports a currency value.

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.

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.

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 Japanese yen, '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.

Region	Cost of Goods	Currency_parms
EMEA	$1,247,925.00	1,247,925.00¥
North America	$1,457,020.00	1,457,020.00¥
Oceania	$9,613.00	9,613.00¥
South America	$235,800.00	235,800.00¥

Alphanumeric Format

You can use alphanumeric format for any value to be interpreted as a sequence of characters and composed of any combination of digits, letters, and other characters.

You can also use alphanumeric fields with date display options to provide limited date support. This use of alphanumeric fields is described in *Alphanumeric and Numeric Formats With Date Display Options* on page 118.

The alphanumeric USAGE type is A. The format of the length specification is n, where n is the maximum number of characters in the field. You can have up to 3968 bytes in an alphanumeric field in a FOCUS file segment, and up to 4096 bytes in an XFOCUS file segment. You can have up to 4095 bytes in a fixed-format sequential data source. You may define the length in the Master File, a DEFINE FILE command, or a COMPUTE command.

For example:

Format	Display
A522	The minutes of today's meeting were submitted...
A2	B3
A24	127-A429-BYQ-49

Standard numeric display options are not available for the alphanumeric data format. However, alphanumeric data can be printed under the control of a pattern that is supplied at run time. For instance, if you are displaying a product code in parts, with each part separated by a "-", include the following in a DEFINE command:

```
PRODCODE/A11 = EDIT (fieldname,'999-999-999') ;
```

where:

fieldname

Is the existing field name, not the newly defined field name.

If the value is 716431014, the PRODCODE appears as 716-431-014. See the manual for more information.

Reference: **Usage Notes for 4K Alphanumeric Fields**

- Long alphanumeric fields cannot be indexed.
- For FOCUS data sources, a segment still has to fit on a 4K page. Thus, the maximum length of an alphanumeric field depends on the length of the other fields within its segment.
- You can print or hold long alphanumeric fields, but you cannot view them online.
- Long alphanumeric fields may be used as keys.

Date Formats

Date format enables you to define a field as a date, then manipulate the field value and display that value in ways appropriate to a date. Using date format, you can:

- Define date components, such as year, quarter, month, day, and day of week, and extract them easily from date fields.

- Sort reports into date sequence, regardless of how the date appears.
- Perform arithmetic with dates and compare dates without resorting to special date-handling functions.
- Refer to dates in a natural way, such as JAN 1 1995, without regard to display or editing formats.
- Automatically validate dates in transactions.

Date Display Options

The date format does not specify type or length. Instead, it specifies date component options (D, W, M, Q, Y, and YY) and display options. These options are shown in the following chart.

Display Option	Meaning	Effect
D	Day	Prints a value from 1 to 31 for the day.
M	Month	Prints a value from 1 to 12 for the month.
Y	Year	Prints a two-digit year.
YY	Four-digit year	Prints a four-digit year.
T	Translate month or day	Prints a three-letter abbreviation for months in uppercase, if M is included in the USAGE specification. Otherwise, it prints day of week.
t	Translate month or day	Functions the same as uppercase T (described above), except that the first letter of the month or day is uppercase and the following letters are lowercase.*
TR	Translate month or day	Functions the same as uppercase T (described above), except that the entire month or day name is printed instead of an abbreviation.

Display Option	Meaning	Effect
`tr`	Translate month or day	Functions the same as lowercase t (described above), except that the entire month or day name is printed instead of an abbreviation.*
`Q`	Quarter	Prints Q1 - Q4.
`W`	Day-of-Week	If it is included in a USAGE specification with other date component options, prints a three-letter abbreviation of the day of the week in uppercase. If it is the only date component option in the USAGE specification, it prints the number of the day of the week (1-7; Mon=1).
`w`	Day-of-Week	Functions the same as uppercase W (described above), except that the first letter is uppercase and the following letters are lowercase.*
`WR`	Day-of-Week	Functions the same as uppercase W (described above), except that the entire day name is printed instead of an abbreviation.
`wr`	Day-of-Week	Functions the same as lowercase w (described above), except that the entire day name is printed instead of an abbreviation.*
`J[UL]or JULIAN`	Julian format	Prints date in Julian format.
`YYJ[UL]`	Julian format	Prints a Julian format date in the format YYYYDDD. The 7-digit format displays the four-digit year and the number of days counting from January 1. For example, January 3, 2001 in Julian format is 2001003.

***Note:** When using these display options, be sure they are actually stored in the Master File as lowercase letters.

The following combinations of date components are not supported in date formats:

I2M, A2M, I2MD, A2MD

Reference: How Field Formats Y, YY, M, and W Are Stored

The Y, YY, and M formats are not smart dates. Smart date formats YMD and YYMD are stored as an offset from the base date of 12/31/1900. Smart date formats YM, YQ, YYM, and YYQ are stored as an offset from the base date 01/1901 on z/OS and 12/1900 on other platforms. W formats are stored as integers with a display length of one, containing values 1-7 representing the days of the week. Y, YY, and M formats are stored as integers. Y and M have display lengths of two. YY has a display length of four. When using Y and YY field formats, keep in mind these two important points:

- The Y formats do not sort based on DEFCENT and YRTHRESH settings. A field with a format of Y does not equal a YY field, as this is not a displacement, but a 4-digit integer.
- It is possible to use DEFCENT and YRTHRESH to convert a field from Y to YY format.

Reference: Date Literals Interpretation Table

This table illustrates the behavior of date formats. The columns indicate the number of input digits for a date format. The rows indicate the usage or format of the field. The intersection of row and column describes the result of input and format.

Date Format	1	2	3	4
YYMD	*	*	CC00/0m/dd	CC00/mm/dd
MDYY	*	*	*	*
DMYY	*	*	*	*
YMD	*	*	CC00/0m/dd	CC00/mm/dd
MDY	*	*	*	*
DMY	*	*	*	*
YYM	CC00/0m	CC00/mm	CC0y/mm	CCyy/mm
MYY	*	*	*	*

Date Format	1	2	3	4
YM	CC00/0m	CC00/mm	CC0y/mm	CCyy/mm
MY	*	*	0m/CCyy	mm/CCyy
M	0m	mm	*	*
YYQ	CC00/q	CC0y/q	CCyy/q	0yyy/q
QYY	*	*	q/CCyy	*
YQ	CC00/q	CC0y/q	CCyy/q	0yyy/q
QY	*	*	q/CCyy	*
Q	q	*	*	*
JUL	00/00d	00/0dd	00/ddd	0y/ddd
YYJUL	CC00/00d	CC00/0dd	CC00/ddd	CC0y/ddd
YY	000y	00yy	0yyy	yyyy
Y	0y	yy	*	*
D	0d	dd	*	*
W	w	*	*	*

Date Format	5	6	7	8
YYMD	CC0y/mm/dd	CCyy/mm/dd	0yyy/mm/dd	yyyy/mm/dd
MDYY	0m/dd/CCyy	mm/dd/CCyy	0m/dd/yyyy	mm/dd/yyyy
DMYY	0d/mm/CCyy	dd/mm/CCyy	0d/mm/yyyy	dd/mm/yyyy
YMD	CC0y/mm/dd	CCyy/mm/dd	0yyy/mm/dd	yyyy/mm/dd

Date Format	5	6	7	8
MDY	0m/dd/CCyy	mm/dd/CCyy	0m/dd/yyyy	mm/dd/yyyy
DMY	0d/mm/CCyy	dd/mm/CCyy	0d/mm/yyyy	dd/mm/yyyy
YYM	0yyy/mm	yyyy/mm	*	*
MYY	0m/yyyy	mm/yyyy	*	*
YM	0yyy/mm	yyyy/mm	*	*
MY	0m/yyyy	mm/yyyy	*	*
M	*	*	*	*
YYQ	yyyy/q	*	*	*
QYY	q/yyyy	*	*	*
YQ	yyyy/q	*	*	*
QY	q/yyyy	*	*	*
Q	*	*	*	*
JUL	yy/ddd	*	*	*
YYJUL	CCyy/ddd	0yyy/ddd	yyyy/ddd	*
YY	*	*	*	*
Y	*	*	*	*
D	*	*	*	*
W	*	*	*	*

Note:

- CC stands for two century digits provided by DFC/YRT settings.
- * stands for message FOC177 (invalid date constant).
- Date literals are read from right to left. Date literals and fields can be used in computational expressions, as described in the manual.

Controlling the Date Separator

You can control the date separators when the date appears. In basic date format, such as YMD and MDYY, the date components appear separated by a slash character (/). The same is true for the year-month format, which appears with the year and quarter separated by a blank (for example, 94 Q3 or Q3 1994). The single component formats display just the single number or name.

The separating character can also be a period, a dash, or a blank, or can even be eliminated entirely. The following table shows the USAGE specifications for changing the separating character.

Format	Display
`YMD`	`93/12/24`
`Y.M.D`	`93.12.24`
`Y-M`	`93-12`
`YBMBD`	`93 12 24` (The letter B signifies blank spaces.)
`Y|M|D`	`931224` (The concatenation symbol (|) eliminates the separation character.)

Note:

- You can change the date separator in the following date formats: YYMD, MDYY, DMYY, YMD, MDY, DMY, YYM, MYY, YM, MY, YYQ, QYY, YQ, and QY.
- You cannot change the date separator in a format that includes date translation options.

- You cannot change the date separator (/) in an alphanumeric or numeric format with date display options (for example, I8YYMD).

Date Translation

Numeric months and days can be replaced by a translation, such as JAN, January, Wed, or Wednesday. The translated month or day can be abbreviated to three characters or fully spelled out. It can appear in either uppercase or lowercase. In addition, the day of the week (for example, Monday) can be appended to the beginning or end of the date. All of these options are independent of each other.

Translation	Display
MT	JAN
Mt	Jan
MTR	JANUARY
Mtr	January
WR	MONDAY
wr	Monday

Example: **Using a Date Format**

The following chart shows sample USAGE and ACTUAL formats for data stored in a non-FOCUS data source. The Value column shows the actual data value, and the Display column shows how the data appears.

USAGE	ACTUAL	Value	Display
wrMtrDYY	A6YMD	990315	Monday, March 15 1999
YQ	A6YMD	990315	99 Q1
QYY	A6YMD	990315	Q1 1999

USAGE	ACTUAL	Value	Display
YMD	A6	990315	99/03/15
MDYY	A6YMD	990315	03/15/1999

Note that the date attributes in the ACTUAL format specify the order in which the date is stored in the non-FOCUS data source. If the ACTUAL format does not specify the order of the month, day, and year, it is inferred from the USAGE format.

Using a Date Field

A field formatted as a date is automatically validated when entered. It can be entered as a natural date literal (for example, JAN 12 1999) or as a numeric date literal (for example, 011299).

Natural date literals enable you to specify a date in a natural, easily understandable way, by including spaces between date components and using abbreviations of month names. For example, April 25, 1999 can be specified as any of the following natural date literals:

```
APR 25 1999
25 APR 1999
1999 APR 25
```

Natural date literals can be used in all date computations, and all methods of data source updating. The following code shows examples:

```
In WHERE screening                    WHERE MYDATE IS 'APR 25 1999'
In arithmetic expressions             MYDATE - '1999 APR 25'
In computational date comparisons     IF MYDATE GT '25 APR 1999'

In comma-delimited data          ...,MYDATE = APR 25 1999, ...
```

The following chart describes the format of natural date literals.

Literal	Format
Year-month-day	Four-digit year, uppercase three-character abbreviation, or uppercase full name, of the month, and one-digit or two-digit day of the month (for example, 1999 APR 25 or APRIL 25 1999).
Year-month	Year and month as described above.

Literal	Format
Year-quarter	Year as described above, Q plus quarter number for the quarter (for example, 1999 Q3).
Month	Month as described above.
Quarter	Quarter as described above.
Day of week	Three-character, uppercase abbreviation, or full, uppercase name, of the day (for example, MON or MONDAY).

The date components of a natural date literal can be specified in any order, regardless of their order in the USAGE specification of the target field. Date components are separated by one or more blanks.

For example, if a USAGE specification for a date field is YM, a natural date literal written to that field can include the year and month in any order. MAY 1999 and 1990 APR are both valid literals.

Numeric Date Literals

Numeric date literals differ from natural date literals in that they are simple strings of digits. The order of the date components in a numeric date literal must match the order of the date components in the corresponding USAGE specification. In addition, the numeric date literal must include all of the date components included in the USAGE specification. For example, if the USAGE specification is DMY, then April 25 1999 must be represented as:

```
250499
```

Numeric date literals can be used in all date computations and all methods of data source updating.

Date Fields in Arithmetic Expressions

The general rule for manipulating date fields in arithmetic expressions is that date fields in the same expression must specify the same date components. The date components can be specified in any order, and display options are ignored. Y or YY, Q, M, W, and D are valid components.

Note that arithmetic expressions assigned to quarters, months, or days of the week are computed modulo 4, 12, and 7, respectively, so that anomalies like fifth quarters and thirteenth months are avoided.

For example, if NEWQUARTER and THISQUARTER both have USAGE specifications of Q, and the value of THISQUARTER is 2, then the following statement gives NEWQUARTER a value of 1 (that is, the remainder of 5 divided by 4):

```
NEWQUARTER = THISQUARTER + 3
```

Converting a Date Field

Two types of conversion are possible: format conversion and date component conversion. In the first case, the value of a date format field can be assigned to an alphanumeric or integer field that uses date display options (see the following section). The reverse conversion is also possible.

In the second case, a field whose USAGE specifies one set of date components can be assigned to another field specifying different date components.

For example, the value of REPORTDATE (DMY) can be assigned to ORDERDATE (Y). In this case, the year is being extracted from REPORTDATE. If REPORTDATE is Apr 27 99, ORDERDATE is 99.

You can also assign the value of ORDERDATE to REPORTDATE. If the value of ORDERDATE is 99, the value of REPORTDATE is Jan 1 99. In this case, REPORTDATE is given values for the missing date components.

Syntax: How to Convert a Date Field

```
field1/format = field2;
```

where:

field1

Is a date format field, or an alphanumeric or integer format field using date display options.

format

Is the USAGE (or FORMAT) specification of *field1* (the target field).

field2

Is a date format field, or an alphanumeric or integer format field using date display options. The format types (alphanumeric, integer, or date) and the date components (YY, Y, Q, M, W, D) of *field1* and *field2* do not need to match.

How a Date Field Is Represented Internally

Date fields are represented internally as four-byte binary integers indicating the time elapsed since the date format base date. For each field, the unit of elapsed time is that field smallest date component.

For example, if the USAGE specification of REPORTDATE is MDY, then elapsed time is measured in days, and internally the field contains the number of days elapsed between the entered date and the base date. If you enter the numeric literal for February 13, 1964 (that is, 021364), and then print the field in a report, 02/13/64 appears. If you use it in the equation:

```
NEWDATE = 'FEB 28 1964' - REPORTDATE ;
DAYS/D = NEWDATE ;
```

then the value of DAYS is 15. However, the internal representation of REPORTDATE is a four byte binary integer representing the number of days between December 31, 1900 and February 13, 1964.

Just as the unit of elapsed time is based on a field smallest date component, so too is the base date. For example, for a YQ field, elapsed time is measured in quarters, and the base date is the first quarter of 1901 on z/OS and the last quarter of 1900 on other platforms. For a YM field, elapsed time is measured in months, and the base date is the first month of 1901 on z/OS and the last month of 1900 on other platforms.

To display blanks or the actual base date in a report, use the SET DATEDISPLAY command described in the manual. The default value, OFF, displays blanks when a date matches the base date. ON displays the actual base date value.

You do not need to be concerned with the date format internal representation, except to note that all dates set to the base date appear as blanks, and all date fields that are entered blank or as all zeroes are accepted during validation and interpreted as the base date. They appear as blanks, but are interpreted as the base date in date computations and expressions.

There are several types of formats you can use to represent date components, and the different types do not represent the same values or offsets.

- Full date formats are stored as the number of days from the base date 12/31/1900.
- Single component formats Y, YY, M, W, and D are stored as integers, not as offsets from a base date.
- Partial date formats YM, YQ, YYM, and YYQ are stored as an offset from the base date 01/1901 on z/OS and 12/1900 on other platforms. Both of these base dates are different from the base date for full component dates (12/31/1900). The offset is expressed in a number of months from the base date for YM and YYM, and as a number of quarters from the base date for YQ and YYQ.

- Full date formats with component display options, such as YYMDy, YYMDm, and YYMDq, display the same values as the component formats such as YYQ and YM. However, they are stored as full component dates (the number of days from the base date 12/31/1900). Therefore, while they display the same types of dates as the component date formats YM, YYM, YQ, and YYQ, their internal offset values are not the same. They are considered different types of date formats and cannot be compared or subtracted.
- When either a partial date or a component date is assigned to a field with a full date format, the missing components are assigned the value 01.

Example: Using Full Date Formats and Component Date Formats

The following request retrieves the current date as a full date field named FULLDATE. It creates a partial date field named PARTIALDATE with format YYM, and a component date field named FULLCOMPONENT with format YYMDm, from the full date field. It then creates two new full dates FULLDATE2 and FULLDATE3 by assigning the partial date to one and the component date to the other.

```
DEFINE FILE GGSALES
FULLDATE/YYMD = '2017/09/12';
PARTIALDATE/YYM =FULLDATE;
FULLCOMPONENT/YYMDm = FULLDATE;
FULLDATE2/YYMD = FULLCOMPONENT;
FULLDATE3/YYMD = PARTIALDATE;
END
TABLE FILE GGSALES
PRINT FULLDATE PARTIALDATE FULLCOMPONENT FULLDATE2 FULLDATE3
BY CATEGORY
WHERE RECORDLIMIT EQ 1
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
GRID=OFF,$
ENDSTYLE
END
```

The output is shown in the following image. Note that when the partial and component dates are assigned to FULLDATE2 and FULLDATE3, the day assigned is 01 in both cases.

Category	FULLDATE	PARTIALDATE	FULLCOMPONENT	FULLDATE2	FULLDATE3
Coffee	2017/09/12	2017/09	2017/09	2017/09/01	2017/09/01

Displaying a Non-Standard Date Format

By default, if a date field in a non-FOCUS data source contains an invalid date, a message appears and the entire record fails to appear in a report. For example, if a date field contains '980450' with an ACTUAL of A6 and a USAGE of YMD, the record containing that field does not appear. The SET ALLOWCVTERR command enables you to display the rest of the record that contains the incorrect date.

Note: The ALLOWCVTERR parameter is not supported for virtual fields.

Syntax: How to Invoke ALLOWCVTERR

```
SET ALLOWCVTERR = {ON|OFF}
```

where:

`ON`
Enables you to display a field containing an incorrect date.

`OFF`
Generates a diagnostic message if incorrect data is encountered, and does not display the record containing the bad data. OFF is the default value.

When a bad date is encountered, ALLOWCVTERR sets the value of the field to either MISSING or to the base date, depending on whether MISSING=ON.

The following chart shows the results of interaction between DATEDISPLAY and MISSING, assuming ALLOWCVTERR=ON and the presence of a bad date.

	MISSING=OFF	MISSING=ON
DATEDISPLAY=ON	Displays Base Date 19001231 or 1901/1	.
DATEDISPLAY=OFF	Displays Blanks	.

DATEDISPLAY affects only how the base date appears. See the manual for a description of DATEDISPLAY.

Date Format Support

Date format fields are used in special ways with the following facilities:

- **Dialogue Manager.** Amper variables can function as date fields if they are set to natural date literals. For example:

```
-SET &NOW = 'APR 25 1960' ;
-SET &LATER = '1990 25 APR' ;
-SET &DELAY = &LATER - &NOW ;
```

  In this case, the value of &DELAY is the difference between the two dates, measured in days: 10,957.

- **Extract files.** Date fields in SAVB and unformatted HOLD files are stored as four-byte binary integers representing the difference between the field face value and the standard base date. Date fields in SAVE files and formatted HOLD files (for example, USAGE WP) are stored without any display options.

- **GRAPH.** Date fields are not supported as sort fields in ACROSS and BY phrases.

- **FML.** Date fields are not supported within the RECAP statement.

Alphanumeric and Numeric Formats With Date Display Options

In addition to the standard date format, you can also represent a date by using an alphanumeric, integer, or packed-decimal field with date display options (D, M, Y, and T). Note, however, that this does not offer the full date support that is provided by the standard date format.

Alphanumeric and integer fields used with date display options have some date functionality when used with special date functions, as described in the manual.

When representing dates as alphanumeric or integer fields with date display options, you can specify the year, month, and day. If all three of these elements are present, then the date has six digits (or eight if the year is presented as four digits), and the USAGE can be:

Format	Display
I6MDY	04/21/98
I6YMD	98/04/21
P6DMY	21/04/98

Format	Display
I8DMYY	21/04/1998

The number of a month (1 to 12) can be translated to the corresponding month name by adding the letter T to the format, immediately after the M. For instance:

Format	Data	Display
I6MTDY	05/21/98	MAY 21 98
I4MTY	0698	JUN 98
I2MT	07	JUL

If the date has only the month element, a format of I2MT displays the value 4 as APR, for example. This is particularly useful in reports where columns or rows are sorted by month. They then appear in correct calendar order. For example, JAN, FEB, MAR, because the sorting is based on the numeric, not alphabetical, values. (Note that without the T display option, I2M is interpreted as an integer with a floating dollar sign.)

Date-Time Formats

The date-time data type supports both the date and time, similar to the timestamp data types available in many relational data sources.

Date-time fields are stored in eight, ten, or 12 bytes: four bytes for date and either four, six, or eight bytes for time, depending on whether the format specifies a microsecond or nanosecond.

Computations only allow direct assignment within data types: alpha to alpha, numeric to numeric, date to date, and date-time to date-time. All other operations are accomplished through a set of date-time functions. See the manual for information on subroutines for manipulating date-time fields.

Date-time formats can also produce output values and accept input values that are compatible with the ISO 8601:2000 date-time notation standard. A SET parameter and specific formatting options enable this notation.

Syntax: How to Enable ISO Standard Date-Time Notation

```
SET DTSTANDARD = {OFF|ON|STANDARD|STANDARDU}
```

where:

OFF

Does not provide compatibility with the ISO 8601:2000 date-time notation standard. OFF is the default value.

ON|STANDARD

Enables recognition and output of the ISO standard formats, including use of T as the delimiter between date and time, use of period or comma as the delimiter of fractional seconds, use of Z at the end of universal times, and acceptance of inputs with time zone information. STANDARD is a synonym for ON.

STANDARDU

Enables ISO standard formats (like STANDARD) and also, where possible, converts input strings to the equivalent universal time (formerly known as Greenwich Mean Time), thus enabling applications to store all date-time values in a consistent way.

Example: Using SET DTSTANDARD

The following request displays date-time values input in ISO 8601:2000 date-time standard formats. With SET DTSTANDARD=OFF, the request terminates with a (FOC177): INVALID DATE CONSTANT:

```
SET DTSTANDARD = &STAND
DEFINE FILE EMPLOYEE
-* The following input is format YYYY-MM-DDThh:mm:ss.sTZD
DT1/HYYMDs  =     DT(2004-06-01T19:20:30.45+01:00);
-* The following input has comma as the decimal separator
DT2/HYYMDs  =     DT(2004-06-01T19:20:30,45+01:00);
DT3/HYYMDs  =     DT(20040601T19:20:30,45);
DT4/HYYMDUs =     DT(2004-06-01T19:20:30,45+01:00);
END
TABLE FILE EMPLOYEE
HEADING CENTER
"DTSANDARD = &STAND "
" "
SUM CURR_SAL NOPRINT DT1 AS 'DT1: INPUT = 2004-06-01T19:20:30.45+01:00'
OVER DT2 AS 'DT2: INPUT = 2004-06-01T19:20:30,45+01:00'
OVER DT3 AS 'DT3: INPUT = 20040601T19:20:30,45'
OVER DT4 AS 'DT4: OUTPUT  FORMAT HYYMDUs'
END
```

With DTSTANDARD= STANDARD, the output shows that the input values were accepted, but the time zone offsets in DT1, DT2, and DT4 (+01:00) were ignored on output. The character U in the format for DT4 causes the T separator to be used between the date and the time:

```
                        DTSANDARD = STANDARD

DT1: INPUT = 2004-06-01T19:20:30.45+01:00  2004-06-01 19:20:30.450
DT2: INPUT = 2004-06-01T19:20:30,45+01:00  2004-06-01 19:20:30.450
DT3: INPUT = 20040601T19:20:30,45          2004-06-01 19:20:30.450
DT4: OUTPUT  FORMAT HYYMDUs                2004-06-01T19:20:30.450
```

With DTSTANDARD= STANDARDU, the output shows that the values DT1, DT2, and DT4 were converted to universal time by subtracting the time zone offsets (+01:00):

```
                        DTSANDARD = STANDARDU

DT1: INPUT = 2004-06-01T19:20:30.45+01:00  2004-06-01 18:20:30.450
DT2: INPUT = 2004-06-01T19:20:30,45+01:00  2004-06-01 18:20:30.450
DT3: INPUT = 20040601T19:20:30,45          2004-06-01 19:20:30.450
DT4: OUTPUT  FORMAT HYYMDUs                2004-06-01T18:20:30.450
```

Describing a Date-Time Field

In a Master File, the USAGE (or FORMAT) attribute determines how date-time field values appear in report output and forms, and how they behave in expressions and functions. For FOCUS data sources, it also determines how they are stored.

Format type H describes date-time fields. The USAGE attribute for a date-time field contains the H format code and can identify either the length of the field or the relevant date-time display options.

The MISSING attribute for date-time fields can be ON or OFF. If it is OFF, and the date-time field has no value, it defaults to blank.

Syntax: How to Describe a Numeric Date-Time Value Without Display Options

This format is appropriate for alphanumeric HOLD files or transaction files.

```
USAGE = Hn
```

where:

n

Is the field length, from 1 to 23, including up to eight characters for displaying the date and up to nine, 12, or 15 characters for the time. For lengths less than 20, the date is truncated on the right.

An eight-character date includes four digits for the year, two for the month, and two for the day of the month, YYYYMMDD.

A nine-character time includes two digits for the hour, two for the minute, two for the second, and three for the millisecond, HHMMSSsss. The millisecond component represents the decimal portion of the second to three places.

A twelve-character time includes two digits for the hour, two for the minute, two for the second, three for the millisecond, and three for the microsecond, HHMMSSsssmmm. The millisecond component represents the decimal portion of the second value to three places. The microsecond component represents three additional decimal places beyond the millisecond value.

A fifteen-character time includes two digits for the hour, two for the minute, two for the second, three for the millisecond, three for the microsecond and three for the nanosecond, HHMMSSsssmmmnnn. The millisecond component represents the decimal portion of the second value to three places. The microsecond component represents three additional decimal places beyond the millisecond value. The nanosecond component represents three additional decimal places beyond the microsecond value.

With this format, there are no spaces between the date and time components, no decimal points, and no spaces or separator characters within either component. The time must be entered using the 24-hour system. For example, the value 19991231225725333444 represents 1999/12/31 10:57:25.333444PM.

Syntax: How to Describe a Time-Only Value

```
USAGE = H timefmt1
```

where:

timefmt1

Is the USAGE format for displaying time only. Hour, minute, and second components are always separated by colons (:), with no intervening blanks. A time value can have a blank immediately preceding an am/pm indicator. For information, see *Display Options for a Time-Only Value* on page 123.

Reference: **Display Options for a Time-Only Value**

The following table lists the valid time display options for a time-only USAGE attribute. Assume the time value is 2:05:27.123456444 a.m.

Option	Meaning	Effect
`H`	Hour (two digits). If the format includes the option a, b, A, or B, the hour value is from 01 to 12. Otherwise, the hour value is from 00 to 23, with 00 representing midnight.	Prints a two-digit hour. For example: `USAGE = HH` prints `02`
`h`	Hour with zero suppression. If the format includes the option a, b, A, or B, the hour value is from 1 to 12. Otherwise, the hour is from 0 to 23.	Displays the hour with zero suppression. For example: `USAGE = Hh` prints `2`
`I`	Minute (two digits). The minute value is from 00 to 59.	Prints the two-digit minute. For example: `USAGE = HHI` prints `02:05`
`i`	Minute with zero suppression. The minute value is from 0 to 59.	Prints the minute with zero suppression. This cannot be used together with an hour format (H or h). For example: `USAGE = Hi` prints `5`
`S`	Second (two digits). S: 00 to 59	Prints the two-digit second. For example: `USAGE = HHIS` prints `02:05:27`

Option	Meaning	Effect
s	Millisecond (three digits, after the decimal point in the second). 000 to 999	Prints the second to three decimal places. For example: `USAGE = HHISs` prints `02:05:27.123`
m	Microsecond (three additional digits after the millisecond). 000 through 999	Prints the second to six decimal places. For example: `USAGE = HSsm` prints `27.123456`
n	Nanosecond (three additional digits after the microsecond). 000 through 999	Prints the second to nine decimal places. For example: `USAGE = HSsn` prints `27.123456444`
x	Instead of using S, s, m, or n, you can specify up to nine decimal places for seconds using the *x* option, where *x* is a number from 1 to 9. Alternatively, you can use the s, m, and n formats to display three, six, or nine decimal places.	`USAGE = HHI1` prints `02:05:27.1`
A	12-hour time display with AM or PM in uppercase.	Prints the hour from 01 to 12, followed by AM or PM. For example: `USAGE = HHISA` prints `02:05:27AM`
a	12-hour time display with am or pm in lowercase.	Prints the hour from 01 to 12, followed by am or pm. For example: `USAGE = HHISa` prints `02:05:27am`
B	12-hour time display with AM or PM in uppercase, with a blank space before the AM or PM.	Prints the hour from 01 to 12, followed by a space and then AM or PM. For example: `USAGE = HHISB` prints `02:05:27 AM`

Option	Meaning	Effect
b	12-hour time display with am or pm in lowercase, with a blank space before the am or pm.	Prints the hour from 01 to 12, followed by a space followed by am or pm. For example: `USAGE = HHISb` prints `02:05:27 am`
Z	24-hour time display with Z to indicate universal time. Z is incompatible with AM/PM output.	Prints the hour from 01 to 24, followed by Z. For example: `USAGE = HHISZ` prints `14:30[:20.99]Z`

When the format includes more than one time display option:

- The options must appear in the order hour, minute, second, millisecond, microsecond, nanosecond.
- The first option must be either hour, minute, or second.
- No intermediate component can be skipped. If hour is specified, the next option must be minute. It cannot be second.

Note: Unless you specify one of the AM/PM time display options, the time component appears using the 24-hour system.

Syntax: How to Describe a Date-Time Value

```
USAGE = Hdatefmt [separator] [timefmt2]
```

where:

datefmt

Is the USAGE format for displaying the date portion of the date-time field. For information, see *Display Options for the Date Component of a Date-Time Field* on page 126.

separator

Is a separator between the date components. The default separator is a slash (/). Other valid separators are: period (.), hyphen (-), blank (B), or none (N). With translated months, these separators can only be specified when the k option is not used.

With the STANDARD and STANDARDU settings, the separator for dates is always hyphen. The separator between date and time is blank by default. However, if you specify the character U as the separator option, the date and time will be separated by the character T.

timefmt2

Is the format for a time that follows a date. Time is separated from the date by a blank. Time components are separated from each other by colons. Unlike the format for time alone, a time format that follows a date format consists of at most two characters: a single character to represent all of the time components that appear and, optionally, one character for an AM/PM option. For information, see *Display Options for the Time Component of a Date-Time Field* on page 129.

Reference: **Display Options for the Date Component of a Date-Time Field**

The date format can include the following display options, as long as they conform to the allowed combinations. In the following table, assume the date is February 5, 1999.

Option	Meaning	Example
`Y`	2-digit year	`99`
`YY`	4-digit year	`1999`
`M`	2-digit month (01 - 12)	`02`
`MT`	Full month name	`February`
`Mt`	Short month name	`Feb`
`D`	2-digit day	`05`
`d`	Zero-suppressed day. A blank space replaces the zero.	` 5`
`e`	Zero-removed day. The day number is shifted to the left, and any components to the right of this are shifted to the left. Requires a date separator.	`5`
`o`	Zero-removed month. Automatically implements the e option for a zero-removed day. The month and day numbers are shifted to the left, and any components to the right of these are also shifted. Required a date separator.	`5`

Option	Meaning	Example

k	For formats in which month or day is followed by year, and month is translated to a short or full name, k separates the year from the day with a comma and blank. Otherwise, the separator is a blank.	`USAGE = HMtDkYY`
		
		`prints Feb 05, 1999`

Note: Unless you specify one of the AM/PM time display options, the time component uses the 24-hour system.

Example: **Using Zero Removal for Date-Time Month and Day Numbers**

The following request creates the date-time value 01/01/2013. It then displays this value using:

- Normal month and day numbers, format HMDYY.
- Month removal and day removal, format HoeYY.
- Month removal without day removal (which forces day removal), format HodYY.
- Day removal without month removal, format HMeYY. Note that month removal is not forced by day removal.

```
DEFINE FILE GGSALES
DATE1A/HMDYY = DT(01/01/2013);
DATE1B/HoeYY = DATE1A;
DATE1C/HodYY = DATE1A;
DATE1D/HMeYY = DATE1A;
END
TABLE FILE GGSALES
SUM DOLLARS NOPRINT
DATE1A AS 'HMDYY'
DATE1B AS 'HoeYY'
DATE1C AS 'HodYY'
DATE1D AS 'HMeYY'
ON TABLE SET PAGE NOPAGE
END
```

The output is:

```
HMDYY       HoeYY       HodYY       HMeYY
-----       -----       -----       -----
01/01/2013  1/1/2013    1/1/2013    01/1/2013
```

Example: Comparing Zero Suppression With Zero Removal

The following request creates two dates with date-time formats in which the date component has a leading zero (01). In the first date, the day component is the first component and displays on the left. In the second date, the day component is the second component and displays in the middle. The request prints these dates:

- With all zeros displayed, format HDMYY.
- With zero suppression for the day component, format HdMYY.
- With zero removal for the day component, format HeMYY.

```
DEFINE FILE GGSALES
DATE1A/HDMYY = DT(01/12/2012);
DATE2A/HMDYY = DT(12/01/2012);
DATE1B/HdMYY = DATE1A;
DATE2B/HMdYY = DATE2A;
DATE1C/HeMYY = DATE1A;
DATE2C/HMeYY = DATE2A;
END
TABLE FILE GGSALES
SUM DOLLARS NOPRINT
DATE1A AS 'HDMYY'
DATE2A AS ''      OVER
DATE1B AS 'HdMYY'
DATE2B AS ''      OVER
DATE1C AS 'HeMYY'
DATE2C AS ''
ON TABLE SET PAGE NOPAGE
```

On the output, the first row shows the date with all zeros displayed. The second row shows zero suppression of the day number, where the zero has been replaced by a blank space so that all the components are aligned with the components on row 1. The last row shows zero removal, where the zero has been removed from the day number, and all of the remaining characters have been shifted over to the left:

```
HDMYY  01/12/2012    12/01/2012
HdMYY   1/12/2012    12/ 1/2012
HeMYY  1/12/2012     12/1/2012
```

Reference: **Display Options for the Time Component of a Date-Time Field**

The following table lists the valid options. Assume the date is February 5, 1999 and the time is 02:05:25.444555333 a.m.

Option	Meaning	Example
`H`	Prints hour.	`USAGE = HYYMDH` prints `1999/02/05 02`
`I`	Prints hour:minute.	`USAGE = HYYMDI` prints `1999/02/05 02:05`
`S`	Prints hour:minute:second.	`USAGE = HYYMDS` prints `1999/02/05 02:05:25`
`s`	Prints hour:minute:second.millisecond.	`USAGE = HYYMDs` prints `1999/02/05 02:05:25.444`
`m`	Prints hour:minute:second.microsecond.	`USAGE = HYYMDm` prints `1999/02/05 02:05:25.444555`
`n`	Prints hour:minute:second.nanosecond.	`USAGE = HYYMDn` prints `1999/02/05 02:05:25.444555333`
x	Instead of using S, s, m, or n, you can specify up to nine decimal places for seconds using the *x* option, where *x* is a number from 1 to 9. Alternatively, you can use the s, m, and n formats to display three, six, or nine decimal places.	`USAGE = HYYMD1` prints `1999/02/05 02:05:25.4`
`A`	Prints AM or PM. This uses the 12-hour system and causes the hour to be printed with zero suppression.	`USAGE = HYYMDSA` prints `1999/02/05 2:05:25AM`
`a`	Prints am or pm. This uses the 12-hour system and causes the hour to be printed with zero suppression.	`USAGE = HYYMDSa` prints `1999/02/05 2:05:25am`

Option	Meaning	Example
B	Prints AM or PM, preceded by a blank space. This uses the 12-hour system and causes the hour to be printed with zero suppression.	`USAGE = HYYMDSB` prints `1999/02/05 2:05:25 AM`
b	Prints am or pm, preceded by a blank space. This uses the 12-hour system and causes the hour to be printed with zero suppression.	`USAGE = HYYMDSb` prints `1999/02/05 2:05:25 am`
Z	Prints Z to indicate universal time. This uses the 24-hour system. Z is incompatible with AM/PM output.	`USAGE = HHISZ` prints `14:30[:20.99]Z`

The date components can be in any of the following combinations and order:

- Year-first combinations: Y, YY, YM, YYM, YMD, YYMD.
- Month-first combinations: M, MD, MY, MYY, MDY, MDYY.
- Day-first combinations: D, DM, DMY, DMYY.

Reference: Date-Time Usage Notes

- In order to have a time component, you must have a day component.
- If you use the k option, you cannot change the date separator.

Character Format A*n*V

The character format A*n*V is supported in Master Files for FOCUS, XFOCUS, and relational data sources. This format is used to represent the VARCHAR (variable length character) data types supported by relational database management systems.

For relational data sources, A*n*V keeps track of the actual length of a VARCHAR column. This information is important when the value is used to populate a VARCHAR column in a different RDBMS. It affects whether trailing blanks are retained in string concatenation and, for Oracle, string comparisons (the other relational engines ignore trailing blanks in string comparisons).

In a FOCUS or XFOCUS data source, A*n*V does not provide true variable length character support. It is a fixed-length character field with two extra leading bytes to contain the actual length of the data stored in the field. This length is stored as a short integer value occupying two bytes. Trailing blanks entered as part of an AnV field count in its length.

Note: Because of the two bytes of overhead and the additional processing required to strip them, A*n*V format is not recommended for use in non-relational data sources.

Syntax: How to Specify A*n*V Fields in a Master File

```
FIELD=name, ALIAS=alias, USAGE=AnV [,ACTUAL=AnV] , $
```

where:

n

Is the size (maximum length) of the field. It can be from 1 to 4093. Note that because of the additional two bytes used to store the length, an A4093V field is actually 4095 bytes long. A size of zero (A0V) is not supported. The length of an instance of the field can be zero.

Note: HOLD FORMAT ALPHA creates an ACTUAL format of A*n*W in the Master File. See *Propagating an AnV Field to a HOLD File* on page 132.

Example: Specifying the A*n*V Format in a Master File

The following represents a VARCHAR field in a Master File for a Db2 data source with size 200:

```
$ VARCHAR FIELD USING AnV
  FIELD=VARCHAR200, ALIAS=VC200, USAGE=A200V, ACTUAL=A200V,MISSING=ON ,$
```

The following represents an A*n*V field in a Master File for a FOCUS data source with size 200:

```
FIELD=ALPHAV, ALIAS=AV200, USAGE=A200V, MISSING=ON ,$
```

If a data source has an A*n*V field, specify the following in order to create a HOLD FORMAT ALPHA file without the length designator:

```
FIELD=ALPHA, USAGE=A25, ACTUAL=A25V, $
```

or

```
DEFINE ...
  ALPHA/A25 = VARCHAR ;
END
```

or

```
COMPUTE ALPHA/A25 = VARCHAR ;
```

In order to alter or create a Master File to include A*n*V, the data must be converted and the length added to the beginning of the field. For example, issue a HOLD command when the field is described as follows:

```
FIELD=VARCHAR, ,USAGE=A25V, ACTUAL=A25, $
```

or

```
DEFINE ...
  VARCHAR/A25V = ALPHA ;
END
```

or

```
COMPUTE VARCHAR/A25V = ALPHA ;
```

Reference: Usage Notes for A*n*V Format

- A*n*V can be used anywhere that A*n* can be used, except for the restrictions listed in these notes.
- Full FOCUS and SQL operations are supported with this data type, including CREATE FILE for relational data sources.
- Joins are not supported between A*n* and A*n*V fields.
- DBCS characters are supported. As with the A*n* format, the number of characters must fit within the 4K data area.
- COMPUTE and DEFINE generate the data type specified on the left-hand side.
- Conversion between A*n*V and TX fields is not supported.
- A*n*V fields cannot have date display options.

Reference: Propagating an A*n*V Field to a HOLD File

When a user propagates an A*n*V field to a sequential data source using the HOLD FORMAT ALPHA command, the two-byte integer length is converted to a six-digit alphanumeric length. The field in the HOLD file consists of this six-digit number followed by the character data. The format attributes for this field are:

```
... USAGE=AnV, ACTUAL=AnW
```

A*n*W is created as a by-product of HOLD FORMAT ALPHA. However, it can be read and used for input as necessary. The number of bytes occupied by this field in the HOLD file is 6+*n*.

Example: Propagating an A*n*V Field to a HOLD File

The A39V field named TITLEV, is propagated to the HOLD file as:

```
FIELDNAME = TITLEV ,E03 ,A39V ,A39W ,$
```

In a binary HOLD file, the USAGE and ACTUAL formats are A*n*V, although the ACTUAL format may be padded to a full 4-byte word. The number of bytes occupied by this field in the HOLD file is 2+*n*.

When an A*n*V field is input into a data source, all bytes in the input field beyond the given length are ignored. These bytes are set to blanks as part of the input process.

When a user creates a relational data source using the HOLD FORMAT *sqlengine* command, the A*n*V field generates a VARCHAR column in the relational data source.

For example, the A39V field named TITLEV, is propagated to a HOLD FORMAT DB2 file as:

```
FIELDNAME = 'TITLEV', 'TITLEV', A39V, A39V ,$
```

Text Field Format

You can store any combination of characters as a text field.

Syntax: How to Specify a Text Field in a Master File

```
FIELD = fieldname, ALIAS = aliasname, USAGE = TXn[F],$
```

where:

fieldname
Is the name you assign the text field.

aliasname
Is an alternate name for the field name.

n
Is the output display length in TABLE for the text field. The display length may be between 1 and 256 characters.

All letters, digits, and special characters can be stored with this format. The following are some sample text field formats.

Format	Display
TX50	This course provides the DP professional with the skills needed to create, maintain, and report from FOCUS data sources.
TX35	This course provides the DP professional with the skills needed to create, maintain, and report from FOCUS data sources.

The standard edit options are not available for the text field format.

Reference: **Usage Notes for Text Field Format**

- Conversion between text and alphanumeric fields is supported in DEFINE and COMPUTE commands.
- Multiple text fields are supported, and they and be anywhere in the segment.

The Stored Data Type: ACTUAL

ACTUAL describes the type and length of data as it is actually stored in the data source. While some data types, such as alphanumeric, are universal, others differ between different types of data sources. Some data sources support unique data types. For this reason, the values you can assign to the ACTUAL attribute differ for each type of data source.

ACTUAL Attribute

This attribute describes the type and length of your data as it actually exists in the data source. The source of this information is your existing description of the data source (such as a COBOL FD statement). The ACTUAL attribute is one of the distinguishing characteristics of a Master File for non-FOCUS data sources. Since this attribute exists only to describe the format of a non-FOCUS data structure, it is not used in the Master File of a FOCUS data structure.

If your data source has a date stored as an alphanumeric field and you need to convert it to a date for sorting or aggregation in a report, you can use the DATEPATTERN attribute in the Master File. then uses the pattern specified to convert the alphanumeric date to a date.

Syntax: How to Specify the ACTUAL Attribute

```
ACTUAL = format
```

where:

format
Consists of values taken from the following table, which shows the codes for the types of data that can be read.

ACTUAL Type	Meaning
DATE	Four-byte integer internal format, representing the difference between the date to be entered and the date format base date.
A*n*	Where *n* = 1-4095 for fixed-format sequential and VSAM data sources, and 1-256 for other non-FOCUS data sources. Alphanumeric characters A-Z, 0-9, and the special characters in the EBCDIC display mode. A*n* accepts all the date-time string formats, as well as the H*n* display formats. ACTUAL=A*n* also accepts a date-time field as it occurs in an alphanumeric HOLD file or SAVE file.
D8	Double-precision, floating-point numbers, stored internally in eight bytes.
F4	Single-precision, floating-point numbers, stored internally in four bytes.
H*n*	H8, H10, or H12 accepts a date-time field as it occurs in a binary HOLD file or SAVB file.
I*n*	Binary integers: I1 = single-byte binary integer. I2 = half-word binary integer (2 bytes). I4 = full-word binary integer (4 bytes). I8 = double-word binary integer (8 bytes). **Note:** The USAGE must be P or D. Decimals are honored, with proper conversion to the decimals of the P or D USAGE.

ACTUAL Type	Meaning
P*n*	Where *n* = 1-16. Packed decimal internal format. *n* is the number of bytes, each of which contains two digits, except for the last byte which contains a digit and the sign (+ or -). For example, P6 means 11 digits plus a sign.
Z*n*	Where *n* = 1-31. Zoned decimal internal format. *n* is the number of digits, each of which takes a byte of storage. The last digit contains a digit and the sign. If the field contains an assumed decimal point, represent the field with an ACTUAL format of Z*n* and a USAGE format of P*m*.*d*, where *m* is the total number of digits in the display plus the assumed decimal point, *d* is the number of decimal places, and *m* must be at least 1 greater than the value of *n*. For example, a field with ACTUAL=Z5 and one decimal place needs USAGE=P6.1 (or P7.1, or greater).

Note:

- Unless your data source is created by a program, all of the characters are either of type A (alphanumeric) or type Z (zoned decimal).
- ACTUAL formats supported for date-time values are A*n*, H8, H10, and H12. A*n* accepts all the date-time string formats, as well as the H*n* USAGE display format. ACTUAL=H8, H10, or H12 accepts a date-time field as it occurs in a binary HOLD file or SAVB file. ACTUAL=A*n* accepts a date-time field as it occurs in an alphanumeric HOLD file or SAVE file.
- If you create a binary HOLD file from a data source with a date-time field, the ACTUAL format for that field is of the form H*n*. If you create an alphanumeric HOLD file from a data source with a date-time field, the ACTUAL format for that field is of the form A*n*.

Reference: ACTUAL to USAGE Conversion

The following conversions from ACTUAL format to USAGE (display) format are automatically handled and do not require invoking a function:

ACTUAL	USAGE
A	A, D, F, I, P, date format, date-time format

ACTUAL	USAGE
D	D
DATE	date format
F	F
H	H
I	I, date format
P	P, date format
Z	D, F, I, P

Reference: **COBOL Picture to USAGE Format Conversion**

The following table shows the USAGE and ACTUAL formats for COBOL, FORTRAN, PL1, and Assembler field descriptions.

COBOL USAGE FORMAT	BYTES OF COBOL PICTURE	INTERNAL STORAGE	ACTUAL FORMAT	USAGE FORMAT
DISPLAY DISPLAY DISPLAY DISPLAY	X(4) S99 9(5)V9 99	4 2 6 2	A4 Z2 Z6.1 A2	A4 P3 P8.1 A2
COMP COMP COMP* COMP COMP-1**	S9 S9(4) S9(5) S9(9) —	4 4 4 4 4	I2 I2 I4 I4 F4	I1 I4 I5 I9 F6
COMP-2***	—	8	D8	D15
COMP-3 COMP-3 COMP-3	9 S9V99 9(4)V9(3)	8 8 8	P1 P2 P4	P1 P5.2 P8.3

COBOL USAGE FORMAT	BYTES OF COBOL PICTURE	INTERNAL STORAGE	ACTUAL FORMAT	USAGE FORMAT
FIXED BINARY(7) (COMP-4)	B or XL1	8	I4	I7

* Equivalent to INTEGER in FORTRAN, FIXED BINARY(31) in PL/1, and F in Assembler.

** Equivalent to REAL in FORTRAN, FLOAT(6) in PL/1, and E in Assembler.

*** Equivalent to DOUBLE PRECISION or REAL*8 in FORTRAN, FLOAT(16) in PL/1, and D in Assembler.

Note:

1. The USAGE lengths shown are minimum values. They may be larger if desired. Additional edit options may also be added.
2. In USAGE formats, an extra character position is required for the minus sign if negative values are expected.
3. PICTURE clauses are not permitted for internal floating-point items.
4. USAGE length should allow for maximum possible number of digits.
5. In USAGE formats, an extra character position is required for the decimal point.

Note that FOCUS data sources do not use the ACTUAL attribute, and instead rely upon the USAGE attribute to specify both how a field is stored and formatted.

Null or MISSING Values: MISSING

If a segment instance exists but no data has been entered into one of its fields, that field has no value. Some types of data sources represent this absence of data as a blank space ( ) or zero (0), but others explicitly indicate an absence of data with a null indicator or as a special null value. Null values (sometimes known as missing data) are significant in reporting applications, especially those that perform aggregating functions, such as averaging.

If your type of data source supports missing data, as do FOCUS data sources and most relational data sources, then you can use the optional MISSING attribute to enable null values to be entered into and read from a field. MISSING plays a role when you:

- **Create new segment instances.** If no value is supplied for a field for which MISSING has been turned ON in the Master File or in a DEFINE or COMPUTE definition, then the field is assigned a missing value.

- **Generate reports.** If a field with a null value is retrieved, the field value is not used in aggregating calculations, such as averaging and summing. If the report calls for the field value to display, a special character appears to indicate a missing value. The default character is a period (.), but you can change it to any character string you wish using the SET NODATA command or the SET HNODATA command for HOLD files, as described in the manual.

Syntax: How to Specify a Missing Value

```
MISSING = {ON|OFF}
```

where:

`ON`

Distinguishes a missing value from an intentionally entered blank or zero when creating new segment instances and reporting.

`OFF`

Does not distinguish between missing values and blank or zero values when creating new segment instances and reporting. OFF is the default value.

Reference: Usage Notes for MISSING

Note the following rules when using MISSING:

- **Alias.** MISSING does not have an alias.
- **Value.** It is recommended that you set the MISSING attribute to match the field predefined null characteristic (whether the characteristic is explicitly set when the data source is created, or set by default). For example, if a relational table column has been created with the ability to accept null data, describe the field with the MISSING attribute set to ON so that its null values are correctly interpreted.

  FOCUS data sources also support MISSING=ON, which assigns sets an internal flag for missing values.
- **Changes.** You can change the MISSING attribute at any time. Note that changing MISSING does not affect the actual stored data values that were entered using the old setting. However, it does affect how that data is interpreted. If null data is entered when MISSING is turned ON, and then MISSING is switched to OFF, the data originally entered as null is interpreted as blanks (for alphanumeric fields) or zeroes (for numeric fields).

Using a Missing Value

Consider the field values shown in the following four records:

		1	3

If you average these values without declaring the field with the MISSING attribute, a value of zero is automatically be supplied for the two blank records. Thus, the average of these four records is (0+0+1+3)/4, or 1. If you turn MISSING to ON, the two blank records are not used in the calculation, so the average is (1+3)/2, or 2.

Missing values in a unique segment are also automatically supplied with a zero, a blank, or a missing value depending on the MISSING attribute. What distinguishes missing values in unique segments from other values is that they are not stored. You do have to supply a MISSING attribute for fields in unique segments on which you want to perform counts or averages.

The manual contains a more thorough discussion of using null values (sometimes called missing data) in reports. It includes alternative ways of distinguishing these values in reports, such as using the WHERE phrase with MISSING selection operators, and creating virtual fields using the DEFINE FILE command with the SOME or ALL phrase.

Describing an FML Hierarchy

The Financial Modeling Language (FML) supports dynamic reporting against hierarchical data structures.

You can define the hierarchical relationships between fields in a Master File and automatically display these fields using FML. You can also provide descriptive captions to appear in reports in place of the specified hierarchy field values.

In the Master File, use the PROPERTY=PARENT_OF and REFERENCE=*hierarchyfld* attributes to define the hierarchical relationship between two fields.

The parent and child fields must have the same FORMAT or USAGE, and their relationship should be hierarchical. The formats of the parent and child fields must both be numeric or both alphanumeric.

Syntax: How to Specify a Hierarchy Between Fields in a Master File

```
FIELD=parentfield,...,PROPERTY=PARENT_OF, REFERENCE=[seg.]hierarchyfld,$
```

where:

parentfield

Is the parent field in the hierarchy.

PROPERTY=PARENT_OF

Identifies this field as the parent of the referenced field in a hierarchy.

These attributes can be specified on every field. Therefore, multiple hierarchies can be defined in one Master File. However, an individual field can have only one parent. If multiple fields have PARENT_OF attributes for the same hierarchy field, the first parent found by traversing the structure in top-down, left-to-right order is used as the parent.

seg

Is the segment location of the hierarchy field. Required if more than one segment has a field named *hierarchyfield*.

hierarchyfld

Is the child field in the hierarchy.

PARENT_OF is also allowed on a virtual field in the Master File:

```
DEFINE name/fmt=expression;,PROPERTY=PARENT_OF,REFERENCE=hierarchyfld,$
```

Syntax: How to Assign Descriptive Captions for Hierarchy Field Values

The following attributes specify a caption for a hierarchy field in a Master File

```
FIELD=captionfield,..., PROPERTY=CAPTION, REFERENCE=[seg.]hierarchyfld,$
```

where:

captionfield

Is the name of the field that contains the descriptive text for the hierarchy field. For example, if the employee ID is the hierarchy field, the last name may be the descriptive text that appears on the report in place of the ID.

PROPERTY=CAPTION

Signifies that this field contains a descriptive caption that appears in place of the hierarchy field values.

A caption can be specified for every field, but an individual field can have only one caption. If multiple fields have CAPTION attributes for the same hierarchy field, the first parent found by traversing the structure in top-down, left-to-right order is used as the caption.

seg

Is the segment location of the hierarchy field. Required if more than one segment has a field named *hierarchyfield*.

hierarchyfld

Is the hierarchy field.

CAPTION is also allowed on a virtual field in the Master File:

```
DEFINE name/format=expression;,PROPERTY=CAPTION,REFERENCE=hierarchyfld,$
```

Example: **Defining a Hierarchy in a Master File**

The CENTGL Master File contains a chart of accounts hierarchy. The field GL_ACCOUNT_PARENT is the parent field in the hierarchy. The field GL_ACCOUNT is the hierarchy field. The field GL_ACCOUNT_CAPTION can be used as the descriptive caption for the hierarchy field:

```
FILE=CENTGL     ,SUFFIX=FOC
SEGNAME=ACCOUNTS,SEGTYPE=S01
FIELDNAME=GL_ACCOUNT,           ALIAS=GLACCT,  FORMAT=A7,
          TITLE='Ledger,Account', FIELDTYPE=I, $
FIELDNAME=GL_ACCOUNT_PARENT,    ALIAS=GLPAR,   FORMAT=A7,
          TITLE=Parent,
          PROPERTY=PARENT_OF, REFERENCE=GL_ACCOUNT, $
FIELDNAME=GL_ACCOUNT_TYPE,      ALIAS=GLTYPE,  FORMAT=A1,
          TITLE=Type,$
FIELDNAME=GL_ROLLUP_OP,         ALIAS=GLROLL,  FORMAT=A1,
          TITLE=Op, $
FIELDNAME=GL_ACCOUNT_LEVEL,     ALIAS=GLLEVEL, FORMAT=I3,
          TITLE=Lev, $
FIELDNAME=GL_ACCOUNT_CAPTION,   ALIAS=GLCAP,   FORMAT=A30,
          TITLE=Caption,
          PROPERTY=CAPTION,   REFERENCE=GL_ACCOUNT, $
FIELDNAME=SYS_ACCOUNT,          ALIAS=ALINE,   FORMAT=A6,
          TITLE='System,Account,Line', MISSING=ON, $
```

Validating Data: ACCEPT

ACCEPT is an optional attribute that you can use to validate data as it is entered into .

The ACCEPT attribute supports the following types of operations:

- ACCEPT = *value1* OR *value2* ...

  This option is used to specify one or more acceptable values.

- ACCEPT = *value1* TO *value2*

  This option is used to specify a range of acceptable values.

- ACCEPT = FIND

  This option is used to validate incoming transaction data against a value from a FOCUS data source when performing maintenance operations on another data source. FIND is only supported for FOCUS data sources and does not apply to OLAP-enabled synonyms. Note also that, in the Maintain environment, FIND is not supported when developing a synonym.

- ACCEPT = DECODE

  This option is used to supply pairs of values for auto amper-prompting. Each pair consists of one value that can be looked up in the data source and a corresponding value for display.

- ACCEPT = FOCEXEC

  This option is used to retrieve lookup and display field values by running a FOCEXEC. Each row in the output must include one value for lookup and a corresponding value for display. These values can be anywhere in the row, in any order. The FOCEXEC can return other columns as well.

- ACCEPT = SYNONYM

  This option is used to look up values in another data source and retrieve a corresponding display value. The lookup field values must exist in both data sources, although they do not need to have matching field names. You supply the name of the synonym, the lookup field name, and the display field name.

Syntax: How to Validate Data

```
ACCEPT = list

ACCEPT = value1 TO value2

ACCEPT = FIND (field [AS name] IN file)

ACCEPT=SYNONYM(lookup_field AS display_field IN lookup_synonym)

ACCEPT=FOCEXEC(lookup_field AS display_field IN lookup_focexec)
```

where:

list
Is a string of acceptable values. The syntax is:

value1 OR *value2* OR *value3...*

For example, ACCEPT = RED OR WHITE OR BLUE. You can also use a blank as an item separator. If the list of acceptable values runs longer than one line, continue it on the next. The list is terminated by a comma.

value1 TO *value2*

Gives the range of acceptable values. For example, ACCEPT = 150 TO 1000.

FIND

Verifies the incoming data against the values in another indexed field. This option is available only for FOCUS data sources.

SYNONYM

Looks up values in another data source and retrieves a corresponding display value. The lookup field values must exist in both data sources, although they do not need to have matching field names. You supply the name of the synonym, the lookup field name and the display field name.

FOCEXEC

Retrieves lookup and corresponding display values by running a FOCEXEC. Each row must return a lookup field value and its corresponding display field value anywhere in the row, in any order. You supply the name of the FOCEXEC, the lookup field name, and the display field name.

lookup_field

Is the field in the lookup_synonym or returned by the lookup_focexec whose value will be used in the filter (WHERE dialogue) or by the amper autoprompt facility that will be compared with the field that has the ACCEPT attribute.

display_field

Is the field in the lookup_synonym or returned by the lookup_focexec, whose value will be displayed for selection in the filter dialogue or amper autoprompt drop-down list.

lookup_synonym

Is the name of the synonym that describes the lookup data.

lookup_focexec

Is the name of the FOCEXEC that returns the lookup and display field values, in any order. This FOCEXEC can return other field values as well.

Any value in the ACCEPT that contains an embedded blank (for example, Great Britain) must be enclosed within single quotation marks.

If the ACCEPT attribute is included in a field declaration and the SET command parameter ACCBLN has a value of OFF, blank ( ) and zero (0) values are accepted only if they are explicitly coded into the ACCEPT. SET ACCBLN is described in the manual.

Example: **Specifying a List With an Embedded Blank**

```
ACCEPT = SPAIN OR ITALY OR FRANCE OR 'GREAT BRITAIN'
```

Reference: **Usage Notes for ACCEPT**

Note the following rules when using ACCEPT:

- **Alias.** ACCEPT does not have an alias.
- **Changes.** You can change the information in an ACCEPT attribute at any time.
- **Virtual fields.** You cannot use the ACCEPT attribute to validate virtual fields created with the DEFINE attribute.
- **HOLD files.** If you wish to propagate the ACCEPT attribute into the Master File of a HOLD file, use the SET HOLDATTR command. HOLD files are discussed in the manual.
- **ACCEPT** is used only in MODIFY procedures. It is useful for providing one central validation list to be used by several procedures. The FIND function is useful when the list of values is large or undergoes frequent change.

Alternative Report Column Titles: TITLE

When you generate a report, each column title in the report defaults to the name of the field that appears in that column. However, you can change the default column title by specifying the optional TITLE attribute for that field.

You can also specify a different column title within an individual report by using the AS phrase in that report request, as described in the manual.

Note that the TITLE attribute has no effect in a report if the field is used with a prefix operator, such as AVE. You can supply an alternative column title for fields used with prefix operators by using the AS phrase.

Master Files support TITLE attributes for multiple languages. For information, see *Multilingual Metadata* on page 147.

Syntax: How to Specify an Alternative Title

```
TITLE = 'text'
```

where:

text

Is any string of up to 512 characters. You can split the text across as many as five separate title lines by separating the lines with a comma (,). Include blanks at the end of a column title by including a slash (/) in the final blank position. You must enclose the string within single quotation marks if it includes commas or leading blanks.

Example: Replacing the Default Column Title

The following FIELD declaration:

```
FIELD = LNAME, ALIAS = LN, USAGE = A15, TITLE = 'Client,Name',$
```

replaces the default column heading, LNAME, with the following:

```
Client
Name
------
```

Reference: Usage Notes for TITLE

Note the following rules when using TITLE:

- **Alias.** TITLE does not have an alias.
- **Changes.** You can change the information in TITLE at any time. You can also override the TITLE with an AS name in a request, or turn it off with the SET TITLES=OFF command.
- **Virtual fields.** If you use the TITLE attribute for a virtual field created with the DEFINE attribute, the semicolon (;) terminating the DEFINE expression must be on the same line as the TITLE keyword.
- **HOLD files.** To propagate the TITLE attribute into the Master File of a HOLD file, use the SET HOLDATTR command. HOLD files are discussed in the manual.

Documenting the Field: DESCRIPTION

DESCRIPTION is an optional attribute that enables you to provide comments and other documentation for a field within the Master File. You can include any comment up to 2K (2048) characters in length.

Note that you can also add documentation to a field declaration, or to a segment or file declaration, by typing a comment in the columns following the terminating dollar sign. You can even create an entire comment line by inserting a new line following a declaration and placing a dollar sign at the beginning of the line.

The DESCRIPTION attribute for a FOCUS data source can be changed at any time without rebuilding the data source.

Master Files support description attributes for multiple languages. For information, see *Multilingual Metadata* on page 147.

Syntax: How to Supply Field Documentation

```
DESC[RIPTION] = text
```

where:

`DESCRIPTION`

Can be shortened to DESC. Abbreviating the keyword has no effect on its function.

`text`

Is any string of up to 2K (2048) characters. If it contains a comma, the string must be enclosed within single quotation marks.

Example: Specifying a DESCRIPTION

The following FIELD declaration provides a DESCRIPTION:

```
FIELD=UNITS,ALIAS=QTY,USAGE=I6, DESC='QUANTITY SOLD, NOT RETURNED',$
```

Reference: Usage Notes for DESCRIPTION

Note the following rules when using the DESCRIPTION attribute:

- **Alias.** The DESCRIPTION attribute has an alias of DEFINITION.
- **Changes.** You can change DESCRIPTION at any time.
- **Virtual fields.** You can use the DESCRIPTION attribute for a virtual field created with the DEFINE attribute.

Multilingual Metadata

Master Files support column headings and descriptions in multiple languages.

The heading or description used depends on the value of the LANG parameter and whether a TITLE_*ln* or DESC_*ln* attribute is specified in the Master File, or a set of translation files exist for the Master File, where *ln* identifies the language to which the column heading or description applies.

In a Master File, column headings are taken from:

1. A heading specified in the report request using the AS phrase.
2. A TITLE attribute in the Master File, if no AS phrase is specified in the request and SET TITLES=ON.
3. The field name specified in the Master File, if no AS phrase or TITLE attribute is specified, or if SET TITLES=OFF.

Syntax: How to Activate the Use of a Language

Issue the following command in a supported profile or in a FOCEXEC:

```
SET LANG = lng
```

or

```
SET LANG = ln
```

where:

lng

Is the three-letter abbreviation for the language.

ln

Is the two-letter ISO language code.

Note: If SET LANG is used in a procedure, its value will override the values set in nlscfg.err or in any profile.

Reference: Activating a Language in the NLS Configuration File

In the configuration file, issue the following command:

```
LANG = lng
```

Reference: Languages and Language Code Abbreviations

Language Name	Two-Letter Language Code	Three-Letter Language Abbreviation
Arabic	ar	ARB
Baltic	lt	BAL
Chinese - Simplified GB	zh	PRC
Chinese - Traditional Big-5	tw	ROC
Czech	cs	CZE
Danish	da	DAN
Dutch	nl	DUT
English - American	en	AME or ENG
English - UK	uk	UKE
Finnish	fi	FIN
French - Canadian	fc	FRE
French - Standard	fr	FRE
German - Austrian	at	GER
German - Standard	de	GER
Greek	el	GRE
Hebrew	iw	HEW
Italian	it	ITA
Japanese - Shift-JIS(cp942) on ascii cp939 on EBCDIC	ja	JPN
Japanese - EUC(cp10942) on ascii (UNIX)	je	JPE

Language Name	Two-Letter Language Code	Three-Letter Language Abbreviation
Korean	ko	KOR
Norwegian	no	NOR
Polish	pl	POL
Portuguese - Brazilian	br	POR
Portuguese - Portugal	pt	POR
Russian	ru	RUS
Spanish	es	SPA
Swedish	sv	SWE
Thai	th	THA
Turkish	tr	TUR

Placing Multilingual Metadata Directly in a Master File

You can place TITLE_*ln* and DESCRIPTION_*ln* attributes directly in the Master file, where *ln* specifies the language code.

Note: You can also create a set of language translation files and include the trans_file attribute at the file level of the Master File. For information on this technique, see *Storing Localized Metadata in Language Files*.

Syntax: How to Specify Multilingual Metadata in a Master File

```
FIELDNAME = field, ...
    .
    .
    .
TITLE= default_column_heading, TITLE_ln = column_heading_for_ln,
    .
    .
    .
DESC= default_desc, DESC_ln = desc_for_ln,
    .
    .
    .
```

where:

field

Is a field in the Master File.

default_column_heading

Is the column heading to use when SET TITLES=ON and either the LANG parameter is set to the default language for the server, or another language is set but the Master File has no corresponding TITLE_*ln* attribute for that field. This column heading is also used if the *ln* value is invalid.

default_desc

Is the description to use when either the LANG parameter is set to the default language for the server, or another language is set but the Master File has no corresponding DESC_*ln* attribute for that field. This description is also used if the *ln* value is invalid.

TITLE_*ln* = *column_heading_for_ln*

Specifies the language for which the column heading applies and the text of the column heading in that language. That column heading is used when SET TITLES=ON, the LANG parameter is set to a non-default language for the server, and the Master File has a corresponding TITLE_*ln* attribute, where *ln* is the two-digit code for the language specified by the LANG parameter Valid values for *ln* are the two-letter ISO 639 language code abbreviations. For information, see *Languages and Language Code Abbreviations* on page 149.

`DESC_ln = desc_for_ln`

Specifies the language for which the description applies and the description text in that language. This description is used when the LANG parameter is set to a non-default language for the server and the Master File has a corresponding DESC_*ln* attribute. Valid values for *ln* are the two-letter ISO 639 language code abbreviations.

Reference: Usage Notes for Multilingual Metadata

- To generate the correct characters, all languages used must be on the code page specified at startup.
- Master Files should be stored using the code page.
- Multilingual descriptions are supported with all fields described in the Master File, including DEFINE and COMPUTE fields.
- If you issue a HOLD command with SET HOLDATTR=ON, only one TITLE attribute is propagated to the HOLD Master File. Its value is the column heading that would have appeared on the report output.

Example: Using Multilingual Descriptions in a Master File

The following Master File for the CENTINV data source specifies French descriptions (DESC_FR) and Spanish descriptions (DESC_ES) as well as default descriptions (DESC) for the PROD_NUM and PRODNAME fields:

```
FILE=CENTINV, SUFFIX=FOC, FDFC=19, FYRT=00
 SEGNAME=INVINFO, SEGTYPE=S1, $
  FIELD=PROD_NUM, ALIAS=PNUM, FORMAT=A4, INDEX=I,
   DESCRIPTION='Product Number'
   DESC='Product Number',
   DESC_ES='Numero de Producto',
   DESC_FR='Nombre de Produit', $
  FIELD=PRODNAME, ALIAS=PNAME, FORMAT=A30,
   WITHIN=PRODCAT,
   DESCRIPTION='Product Name'
   DESC_FR='Nom de Produit',
   DESC_ES='Nombre de Producto', $
  FIELD=QTY_IN_STOCK, ALIAS=QIS, FORMAT=I7,
   DESCRIPTION='Quantity In Stock', $
  FIELD=PRICE, ALIAS=RETAIL, FORMAT=D10.2,
   TITLE='Price:',
   DESCRIPTION=Price, $
```

Example: Using Multilingual Titles in a Request

The following Master File for the CENTINV data source specifies French titles (TITLE_FR) and Spanish titles (TITLE_ES) as well as default titles (TITLE) for the PROD_NUM and PRODNAME fields:

```
FILE=CENTINV, SUFFIX=FOC, FDFC=19, FYRT=00
 SEGNAME=INVINFO, SEGTYPE=S1, $
  FIELD=PROD_NUM, ALIAS=PNUM, FORMAT=A4, INDEX=I,
   TITLE='Product,Number:',
   TITLE_FR='Nombre,de Produit:',
   TITLE_ES='Numero,de Producto:',
   DESCRIPTION='Product Number', $
  FIELD=PRODNAME, ALIAS=PNAME, FORMAT=A30,
   WITHIN=PRODCAT,
   TITLE='Product,Name:',
   TITLE_FR='Nom,de Produit:',
   TITLE_ES='Nombre,de Producto:'
   DESCRIPTION='Product Name', $
  FIELD=QTY_IN_STOCK, ALIAS=QIS, FORMAT=I7,
   TITLE='Quantity,In Stock:',
   DESCRIPTION='Quantity In Stock', $
  FIELD=PRICE, ALIAS=RETAIL, FORMAT=D10.2,
   TITLE='Price:',
   DESCRIPTION=Price, $
```

The default language is English and, by default, SET TITLES=ON. Therefore, the following request, uses the TITLE attributes to produce column headings that are all in English:

```
TABLE FILE CENTINV
PRINT PROD_NUM PRODNAME PRICE
WHERE PRICE LT 200
END
```

The output is:

```
Product  Product
Number:  Name:                                Price:
-------  -------                              ------
1004     2 Hd VCR LCD Menu                    179.00
1008     DVD Upgrade Unit for Cent. VCR       199.00
1026     AR3 35MM Camera 10 X                 129.00
1028     AR2 35MM Camera 8 X                  109.00
1030     QX Portable CD Player                169.00
1032     R5 Micro Digital Tape Recorder        89.00
```

Now, issue the following command to set the language to Spanish and run the same request:

```
SET LANG = SPA
```

The output now displays column headings from the TITLE_ES attributes where they exist (Product Number and Product Name). Where no Spanish title is specified (the Price field), the column heading in the TITLE attribute appears:

```
Numero        Nombre
de Producto:  de Producto:                          Price:
------------  ------------                          ------
1004          2 Hd VCR LCD Menu                     179.00
1008          DVD Upgrade Unit for Cent. VCR        199.00
1026          AR3 35MM Camera 10 X                  129.00
1028          AR2 35MM Camera 8 X                   109.00
1030          QX Portable CD Player                 169.00
1032          R5 Micro Digital Tape Recorder         89.00
```

Storing Localized Metadata in Language Files

If you want to centralize localized column titles, descriptions, and prompts, and apply them to multiple Master Files, you can create a set of translation files and use the TRANS_FILE attribute in a Master File to invoke them.

Syntax: How to Create and Invoke Metadata Translation Files

Translation File Naming Conventions

The translation files have names of the following form:

prefix`lng.lng`

where:

prefix

Is a group of characters prepended to each related translation file.

lng

Is a language code.

For example, if the common prefix is *dt*, the French translation file would be named dtfre.lng, and the English translation file would be named dteng.lng.

Translation File Contents

The *prefix*eng.lng file must contain any title, description, and prompt values that you want translated as they appear in the Master File, whether they are in English or another language:

1. Copy each attribute value from the Master File that you want translated into the default (eng) translation file, and assign it an index number. The index numbers do not need to be consecutive or in order. For example:

   ```
   39 = Product,Category
   ```

2. Add the translations of those attribute values to the translation files for the other languages in which you want to display the metadata. Assign the same index number to the translations. For example, in the French translation file:

```
39 = Produit,Catégorie
```

Identifying the Translation Files to Use For a Master File

To specify that a Master File should use a particular set of translation files, identify the common prefix in the FILE declaration of the Master File:

```
FILENAME=filename,  TRANS_FILE=[path]/prefix, ...
```

where:

filename

Is the name specified in the FILE= attribute.

path

Is the information needed for locating the set of translation files. It can be a full path or an app reference. If there is one set of translation files with the prefix being used and it is on the app path, this can be omitted.

prefix

Is the common prefix for the set of translation files.

Invoking the Translation Files for a Request

1. Make sure the server is using a code page that supports the languages to be used.
2. Set the LANGUAGE parameter to the language in which the metadata should be displayed.
3. Run the request.

Example: **Using Translation Files**

The following request uses the WF_RETAIL_LITE data source:

```
TABLE FILE WF_RETAIL
SUM REVENUE_US
BY PRODUCT_CATEGORY
BY PRODUCT_SUBCATEG
ON TABLE SET PAGE NOPAGE
ON TABLE SET STYLE *
TYPE = TITLE, FONT='Trebuchet MS', $
ENDSTYLE
END
```

The output is:

Product Category	Product Subcategory	Revenue
Accessories	Charger	$1,660.83
	Headphones	$20,657.86
	Universal Remote Controls	$19,498.65
Camcorder	Handheld	$17,138.95
	Professional	$6,798.00
	Standard	$25,831.38
Computers	Smartphone	$18,298.75
Media Player	Blu Ray	$93,983.27
	DVD Players	$23,068.10
	Streaming	$1,563.80
Stereo Systems	Home Theater Systems	$36,198.80
	Receivers	$20,680.02
	Speaker Kits	$39,382.11
	iPod Docking Station	$15,507.78
Televisions	Flat Panel TV	$27,209.29
Video Production	Video Editing	$25,837.37

The Master File contains the following TRANS_FILE attribute:

```
FILENAME=WF_RETAIL_LITE,  TRANS_FILE=_EDAHOME/NLS/dt, ...
```

The TRANS_FILE attribute points to files that start with the characters *dt* that are in the NLS folder under the EDAHOME directory. The following sample shows some of the contents of the default translation file, dteng.lng:

```
1 = Age
2 = Age Range
3 = Age Group
6 = Gender
9 = Discount,Rate
10 = Discount,Price,Multiplier
15 = Country
17 = State
19 = City
31 = Customer,Income Range
32 = Customer,Income Subrange
33 = Households
34 = Number of,Earners
35 = Household,Size
36 = Industry
38 = Occupation
39 = Product,Category
40 = Product,Subcategory
41 = Brand Type
```

The text assigned to each number can be found in one of the Master Files associated with the WF_RETAIL_LITE Master File. WF_RETAIL_LITE is a cluster Master File that references fact and dimension Master Files to create a star schema.

The following sample shows the corresponding contents of the French translation file, dtfre.lng:

```
1 = Age
2 = Tranche d'âge
3 = Groupe d'âge
6 = Sexe
9 = Remise,Taux
10 = Remise,Prix,Multiplicateur
15 = Pays
17 = Département
19 = Ville
31 = Client,Tranche de revenus
32 = Client,Sous-tranche de revenus
33 = Ménages
34 = Nombre de,Salariés
35 = Ménage,Taille
36 = Secteur d'activité
38 = Profession
39 = Produit,Catégorie
40 = Produit,Sous-catégorie
41 = Type de marque
```

When the language is set to French, any text to be displayed that is an exact match to an index number in the dteng.lng file will be substituted with the text for the same index number in the dtfre.lng file.

The following version of the request adds the SET LANG=FRE command. The server code page supports English and French:

```
SET LANG = FRE
TABLE FILE WF_RETAIL
SUM REVENUE_US
BY PRODUCT_CATEGORY
BY PRODUCT_SUBCATEG
ON TABLE SET PAGE NOPAGE
ON TABLE SET STYLE *
TYPE = TITLE, FONT='Trebuchet MS', $
ENDSTYLE
END
```

The output has translated column titles:

Produit Catégorie	Produit Sous-catégorie	Recettes
Accessories	Charger	$1,660.83
	Headphones	$20,657.86
	Universal Remote Controls	$19,498.65
Camcorder	Handheld	$17,138.95
	Professional	$6,798.00
	Standard	$25,831.38
Computers	Smartphone	$18,298.75
Media Player	Blu Ray	$93,983.27
	DVD Players	$23,068.10
	Streaming	$1,563.80
Stereo Systems	Home Theater Systems	$36,198.80
	Receivers	$20,680.02
	Speaker Kits	$39,382.11
	iPod Docking Station	$15,507.78
Televisions	Flat Panel TV	$27,209.29
Video Production	Video Editing	$25,837.37

Describing a Virtual Field: DEFINE

DEFINE is an optional attribute used to create a virtual field for reporting. You can derive the virtual field value from information already in the data source (that is, from permanent fields). Some common uses of virtual data fields include:

- Computing new numeric values that are not on the data record.
- Computing a new string of alphanumeric characters from other strings.
- Classifying data values into ranges or groups.

- Invoking subroutines in calculations.

Virtual fields are available whenever the data source is used for reporting.

Syntax: How to Define a Virtual Field

```
DEFINE fieldname/format [(GEOGRAPHIC_ROLE = georole)]
  [REDEFINES field2] = expression;
  [,TITLE='title',]
  [TITLE_ln='titleln', ... ,]
  [,DESC[CRIPTION]='desc',]
  [DESC_ln='descln', ... ,]$
```

where:

fieldname

Is the name of the virtual field. The name is subject to the same conventions as names assigned using the FIELDNAME attribute. FIELDNAME is described in *The Field Name: FIELDNAME* on page 80.

format

Is the field format. It is specified in the same way as formats assigned using the USAGE attribute, which is described in *The Displayed Data Type: USAGE* on page 89. If you do not specify a format, it defaults to D12.2.

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.
- 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 level1).
- 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.
- USCOUNTY_FIPS. US county FIPS code.
- USSTATE. US state name.
- USSTATE_ABBR. US state abbreviation.
- USSTATE_FIPS. US state FIPS code.

- ZIP3. US 3-digit postal code.
- ZIP5. US 5-digit postal code.

field2

Enables you to redefine or recompute a field whose name exists in more than one segment.

expression

Is a valid expression. The expression must end with a semicolon (;). Expressions are fully described in the manual.

Note that when an IF-THEN phrase is used in the expression of a virtual field, it must include the ELSE phrase.

TITLE='*title*'

Is a column title for the virtual field in the default language.

TITLE_*ln*='*titleln*'

Is a column title for the virtual field in the language specified by the language code *ln*.

DESC[RIPTION]='*desc*'

Is a description for the virtual field in the default language.

DESC_*ln*='*descln*'

Is a description for the virtual field in the language specified by the language code *ln*.

Place each DEFINE attribute after all of the field descriptions for that segment.

Example: Defining a Field

The following shows how to define a field called PROFIT in the segment CARS:

```
SEGMENT = CARS ,SEGTYPE = S1 ,PARENT = CARREC, $
   FIELDNAME = DEALER_COST ,ALIAS = DCOST ,USAGE = D7, $
   FIELDNAME = RETAIL_COST ,ALIAS = RCOST ,USAGE = D7, $
   DEFINE PROFIT/D7 = RETAIL_COST - DEALER_COST; $
```

Reference: Usage Notes for Virtual Fields in a Master File

Note the following rules when using DEFINE:

- **Alias.** DEFINE does not have an alias.
- **Changes.** You can change the virtual field declaration at any time.

- A DEFINE FILE command takes precedence over a DEFINE in the Master with same name.
- If the expression used to derive the virtual field invokes a function, parameter numbers and types are not checked unless the USERFCHK parameter is set to FULL.

Using a Virtual Field

A DEFINE attribute cannot contain qualified field names on the left-hand side of the expression. Use the WITH phrase on the left-hand side to place the defined field in the same segment as any real field you choose. This will determine when the DEFINE expression will be evaluated.

Expressions on the right-hand side of the DEFINE can refer to fields from any segment in the same path. The expression on the right-hand side of a DEFINE statement in a Master File can contain qualified field names.

A DEFINE attribute in a Master File can refer to only fields in its own path. If you want to create a virtual field that derives its value from fields in several different paths, you have to create it with a DEFINE FILE command using an alternate view prior to a report request, as discussed in the manual. The DEFINE FILE command is also helpful when you wish to create a virtual field that is only used once, and you do not want to add a declaration for it to the Master File.

Virtual fields defined in the Master File are available whenever the data source is used, and are treated like other stored fields. Thus, a field defined in the Master File cannot be cleared in your report request.

A virtual field cannot be used for cross-referencing in a join. It can, however, be used as a host field in a join.

Note: Maintain Data does not support DEFINE attributes that have a constant value. Using such a field in a Maintain Data procedure generates the following message:

```
(FOC03605) name is not recognized.
```

Describing a Calculated Value: COMPUTE

COMPUTE commands can be included in Master Files and referenced in subsequent TABLE requests, enabling you to build expressions once and use them in multiple requests.

Syntax: How to Include a COMPUTE Command in a Master File

```
COMPUTE fieldname/fmt [(GEOGRAPHIC_ROLE = georole)]
 =expression;
  [,TITLE='title',]
  [TITLE_ln='titleln', ... ,]
  [,DESC[CRIPTION]='desc',]
  [DESC_ln='descln', ... ,]$
```

where:

fieldname

Is name of the calculated field.

fmt

Is the format and length of the calculated field.

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.
- 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 level1).
- 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.
- USCOUNTY_FIPS. US county FIPS code.
- USSTATE. US state name.
- USSTATE_ABBR. US state abbreviation.
- USSTATE_FIPS. US state FIPS code.
- ZIP3. US 3-digit postal code.
- ZIP5. US 5-digit postal code.

expression

Is the formula for calculating the value of the field.

TITLE='*title*'

Is a column title for the calculated field in the default language.

`TITLE_ln='titleln'`

Is a column title for the calculated field in the language specified by the language code *ln*.

`DESC[CRIPTION]='desc'`

Is a description for the calculated field in the default language.

`DESC_ln='descln'`

Is a description for the calculated field in the language specified by the language code *ln*.

Reference: Usage Notes for COMPUTE in a Master File

In all instances, COMPUTEs in the Master File have the same functionality and limitations as temporary COMPUTEs. Specifically, fields computed in the Master File must follow these rules:

- They cannot be used in JOIN, DEFINE, or ACROSS phrases, or with prefix operators.
- When used as selection criteria, syntax is either IF TOTAL field or WHERE TOTAL field.
- When used as sort fields, syntax is BY TOTAL COMPUTE field.
- To insert a calculated value into a heading or footing, you must reference it prior to the HEADING or FOOTING command.

Note: Maintain Data does not currently support using COMPUTEs in Master Files.

Example: Coding a COMPUTE in the Master File and Accessing the Computed Value

Use standard COMPUTE syntax to add a calculated value to your Master File. You can then access the calculated value by referencing the computed fieldname in subsequent TABLE requests. When used as a verb object, as in the following example, the syntax is SUM (or PRINT) COMPUTE field.

The following is the SALESTES Master File (the SALES FILE modified with an embedded COMPUTE):

```
FILENAME=SALESTES, SUFFIX=FOC,
SEGNAME=STOR_SEG, SEGTYPE=S1,
   FIELDNAME=STORE_CODE,  ALIAS=SNO,  FORMAT=A3,   $
   FIELDNAME=CITY,        ALIAS=CTY,  FORMAT=A15,  $
   FIELDNAME=AREA,        ALIAS=LOC,  FORMAT=A1,   $

SEGNAME=DATE_SEG, PARENT=STOR_SEG, SEGTYPE=SH1,
   FIELDNAME=DATE,        ALIAS=DTE,  FORMAT=A4MD, $

SEGNAME=PRODUCT, PARENT=DATE_SEG, SEGTYPE=S1,
   FIELDNAME=PROD_CODE,     ALIAS=PCODE,   FORMAT=A3,     FIELDTYPE=I, $
   FIELDNAME=UNIT_SOLD,     ALIAS=SOLD,    FORMAT=I5,     $
   FIELDNAME=RETAIL_PRICE,  ALIAS=RP,      FORMAT=D5.2M,  $
   FIELDNAME=DELIVER_AMT,   ALIAS=SHIP,    FORMAT=I5,     $
   FIELDNAME=OPENING_AMT,   ALIAS=INV,     FORMAT=I5,     $
   FIELDNAME=RETURNS,       ALIAS=RTN,     FORMAT=I3,     MISSING=ON, $
   FIELDNAME=DAMAGED,       ALIAS=BAD,     FORMAT=I3,     MISSING=ON, $

   COMPUTE REVENUE/D12.2M=UNIT_SOLD*RETAIL_PRICE;
```

The following TABLE request uses the REVENUE field:

```
TABLE FILE SALESTES
HEADING CENTER
"NEW YORK PROFIT REPORT"
" "
SUM UNIT_SOLD AS 'UNITS,SOLD' RETAIL_PRICE AS 'RETAIL_PRICE'
COMPUTE REVENUE;
BY PROD_CODE AS 'PROD,CODE'
WHERE CITY EQ 'NEW YORK'
END
```

The output is:

```
           NEW YORK PROFIT REPORT

  PROD  UNITS
  CODE  SOLD   RETAIL_PRICE          REVENUE
  ----  ----   ------------          -------
  B10      30          $.85           $25.50
  B17      20         $1.89           $37.80
  B20      15         $1.99           $29.85
  C17      12         $2.09           $25.08
  D12      20         $2.09           $41.80
  E1       30          $.89           $26.70
  E3       35         $1.09           $38.15
```

Describing a Filter: FILTER

Boolean virtual fields (DEFINE fields that evaluate to TRUE or FALSE) can be used as record selection criteria. If the primary purpose of a virtual field is for use in record selection, you can clarify this purpose and organize virtual fields in the Master File by storing the expression using a FILTER declaration rather than a DEFINE. Filters offer the following features:

- They allow you to organize and store popular selection criteria in a Master File, group them in a Business View, and reuse them in multiple requests and tools.
- For some data sources (such as VSAM and ISAM), certain filter expressions can be inserted inline into the WHERE or IF clause, enhancing optimization compared to a Boolean DEFINE.

Syntax: How to Declare a Filter in a Master File

```
FILTER  filtername = expression; [MANDATORY={YES|NO}]
  [, DESC[RIPTION]='desc']
  [, DESC_ln='descln', ... ] ,$
```

where:

filtername

Is the name assigned to the filter. The filter is internally assigned a format of I1, which cannot be changed.

expression

Is a logical expression that evaluates to TRUE (which assigns the value 1 to the filter field) or FALSE (which assigns the value 0 to the filter field). For any other type of expression, the field becomes a standard numeric virtual field in the Master File. Dialogue Manager variables (amper variables) can be used in the filter expression in same way they are used in standard Master File DEFINEs.

MANDATORY={YES|NO}

Specifies whether to apply the filter even if it is not referenced in a request against the synonym. YES applies the filter to all requests against the synonym. NO applies the filter only when it is referenced in a request. NO is the default value.

Note: Unlike a filter created using the FILTER FILE command, which can be toggled ON and OFF, this setting can only be turned off by removing or changing the value in the Master File.

DESC[RIPTION]='*desc*'

Is a description for the sort object in the default language.

```
DESC_ln='descln'
```

Is a description for the sort object in the language specified by the language code *ln*.

Syntax: How to Use a Master File Filter in a Request

```
TABLE FILE filename
   .
   .
   .
{WHERE|IF} expression_using_filters
```

where:

expression_using_filters

Is a logical expression that references a filter. In a WHERE phrase, the logical expression can reference one or more filters and/or virtual fields.

Reference: Usage Notes for Filters in a Master File

- The filter field name is internally assigned a format of I1 which cannot be changed.
- A filter can be used as a standard numeric virtual field anywhere in a report request, except that they are not supported in WHERE TOTAL tests.
- A mandatory filter can be used to force access to a segment (for example, a table in a cluster synonym) that is not referenced in a request.

Example: Defining and Using a Master File Filter

Consider the following filter declaration added to the MOVIES Master File:

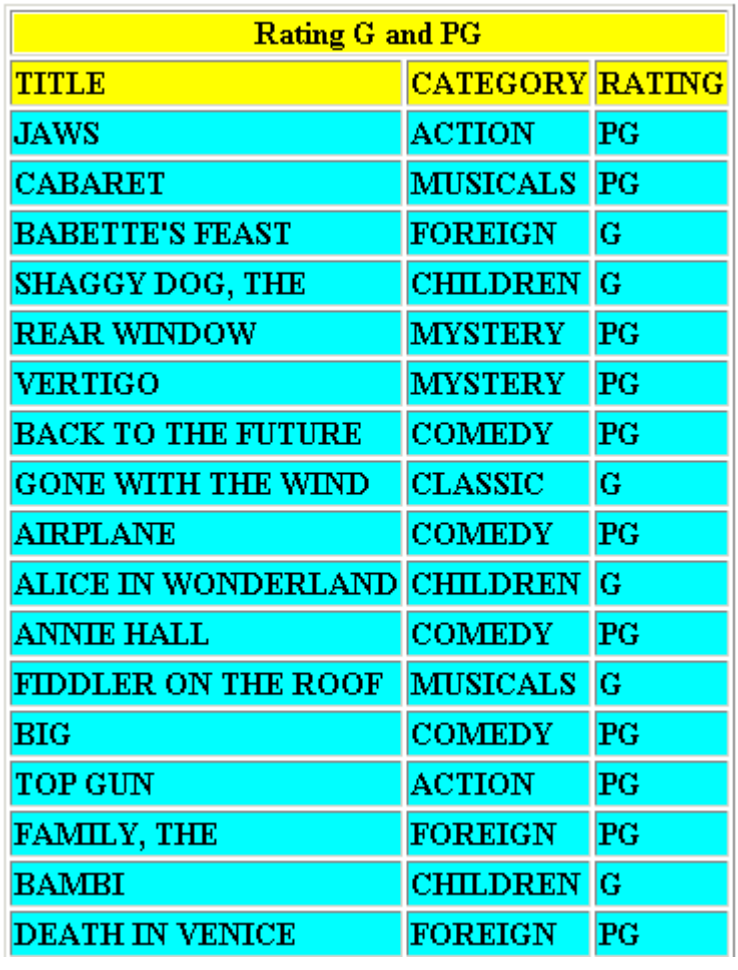
```
FILTER G_RATING = RATING EQ 'G' OR 'PG'; $
```

The following request applies the G_RATING filter:

```
TABLE FILE MOVIES
HEADING CENTER
"Rating G and PG"
PRINT TITLE CATEGORY RATING
WHERE G_RATING
ON TABLE SET PAGE NOPAGE
ON TABLE SET GRID OFF

ON TABLE SET STYLE *
type=report, style=bold, color=black, backcolor=yellow, $
type=data, backcolor=aqua, $
ENDSTYLE
END
```

The output is shown in the following image:



Rating G and PG		
TITLE	CATEGORY	RATING
JAWS	ACTION	PG
CABARET	MUSICALS	PG
BABETTE'S FEAST	FOREIGN	G
SHAGGY DOG, THE	CHILDREN	G
REAR WINDOW	MYSTERY	PG
VERTIGO	MYSTERY	PG
BACK TO THE FUTURE	COMEDY	PG
GONE WITH THE WIND	CLASSIC	G
AIRPLANE	COMEDY	PG
ALICE IN WONDERLAND	CHILDREN	G
ANNIE HALL	COMEDY	PG
FIDDLER ON THE ROOF	MUSICALS	G
BIG	COMEDY	PG
TOP GUN	ACTION	PG
FAMILY, THE	FOREIGN	PG
BAMBI	CHILDREN	G
DEATH IN VENICE	FOREIGN	PG

Example: **Using a Mandatory Filter**

Consider the following filter declaration added to the MOVIES Master File:

```
FILTER G_RATING = RATING EQ 'G' OR 'PG'; MANDATORY=YES ,$
```

The following request does not reference the G_RATING filter:

```
TABLE FILE MOVIES
HEADING CENTER
"Rating G and PG"
PRINT TITLE CATEGORY RATING
ON TABLE SET PAGE NOPAGE
ON TABLE SET GRID OFF

ON TABLE SET STYLE *
type=report, style=bold, color=black, backcolor=yellow, $
type=data, backcolor=aqua, $
ENDSTYLE
END
```

The output is shown in the following image. Note that the G_RATING filter is applied even though it is not referenced in the request:

Rating G and PG		
TITLE	CATEGORY	RATING
JAWS	ACTION	PG
CABARET	MUSICALS	PG
BABETTE'S FEAST	FOREIGN	G
SHAGGY DOG, THE	CHILDREN	G
REAR WINDOW	MYSTERY	PG
VERTIGO	MYSTERY	PG
BACK TO THE FUTURE	COMEDY	PG
GONE WITH THE WIND	CLASSIC	G
AIRPLANE	COMEDY	PG
ALICE IN WONDERLAND	CHILDREN	G
ANNIE HALL	COMEDY	PG
FIDDLER ON THE ROOF	MUSICALS	G
BIG	COMEDY	PG
TOP GUN	ACTION	PG
FAMILY, THE	FOREIGN	PG
BAMBI	CHILDREN	G
DEATH IN VENICE	FOREIGN	PG

Describing a Sort Object: SORTOBJ

You can define sort phrases and attributes in a Master File and reference them by name in a request against the Master File. The entire text of the sort object is substituted at the point in the TABLE where the sort object is referenced. The sort phrases in the sort object are not verified prior to this substitution. The only verification is that there is a sort object name and an equal sign in the Master File SORTOBJ record.

Reference: Usage Notes for Sort Objects in a Master File

- The sort object declaration can appear anywhere after the first SEGNAME/SEGMENT record. However, it must appear after all fields mentioned by it in the Master File, including virtual fields.
- A sort object can use both Master File and local virtual fields.
- Unlimited sort object declarations may appear in a Master File, but the number referenced by a TABLE request cannot result in more than the maximum number of sort phrases in the request.
- The sort object declaration can be followed by optional attributes.
- If a sort object has the same name as a field, the sort object will be used when referenced in a request.

Syntax: How to Declare a Sort Object in a Master File

```
FILE= ...
SEG= ...
FIELD= ...
SORTOBJ sortname = {BY|ACROSS} sortfield1 [attributes]
  [{BY|ACROSS} sortfield2 ... ];
  [,DESC[CRIPTION]='desc',]
  [DESC_ln='descln', ... ,]$
```

where:

sortname

Is a name for the sort object.

sortfield1, sortfield2 ..

Are fields from the Master File or local DEFINE fields that will be used to sort the report output.

attributes

Are any valid sort attributes.

;

Is required syntax for delimiting the end of the sort object expression.

DESC[CRIPTION]='*desc*'

Is a description for the sort object in the default language.

```
DESC_ln='descln'
```

Is a description for the sort object in the language specified by the language code *ln*.

Syntax: How to Reference a Sort Object in a Request

```
TABLE FILE ...
   .
   .
   .
BY sortname   .
   .
   .
END
```

where:

sortname

Is the sort object to be inserted into the request.

Example: Declaring and Referencing a Sort Object

The following sort object for the GGSALES Master File is named CRSORT. It defines two sort phrases:

- BY the REGION field, with a SKIP-LINE attribute.
- ACROSS the CATEGORY field.

```
SORTOBJ CRSORT = ACROSS CATEGORY BY REGION SKIP-LINE ; ,$
```

The following request references the CRSORT sort object:

```
TABLE FILE GGSALES
SUM DOLLARS
BY CRSORT
ON TABLE SET PAGE NOPAGE
END
```

The output is:

```
                    Category
Region       Coffee        Food          Gifts
-----------------------------------------------
Midwest      4178513       4404483       2931349
Northeast    4201057       4445197       2848289
Southeast    4435134       4308731       3037420
West         4493483       4204333       2977092
```

Calling a DEFINE FUNCTION in a Master File

You can reference a DEFINE FUNCTION in an expression in a Master File DEFINE, COMPUTE, or FILTER field. The DEFINE FUNCTION will be loaded into memory when its associated expression is used in a request.

Note: A DEFINE FUNCTION cannot be used in a multi-root Master File.

Syntax: How to Call a DEFINE FUNCTION in a Master File Expression

```
DF.[appname/]filename.functionname(parm1, parm2, ...);
  [DESCRIPTION='description',$
```

where:

appname

Is an optional application name under which the DEFINE FUNCTION FOCEXEC is stored.

filename

Is the name of the FOCEXEC that contains the DEFINE FUNCTION definition. The FOCEXEC can contain multiple DEFINE FUNCTION definitions.

functionname(*parm1, parm2,...*)

Is the function name with the parameters to be used in the expression.

'*description*'

Is an optional description enclosed in single quotation marks.

Example: Using a DEFINE FUNCTION in a Master File

The following DEFINE FUNCTION is stored in the DMFUNCS FOCEXEC. Given a last name and first name, it generates a full name in the format Lastname, Firstname:

```
DEFINE FUNCTION DMPROPER
DESCRIPTION 'Convert name to proper case and last, first format‘
(LASTNAME/A17, FIRSTNAME/A14)
DMPROPER/A34V=LCWORD(17, LASTNAME, 'A17')
             || (', ' | LCWORD(14, FIRSTNAME, 'A14'));
END
```

The following is the DEFINE field named WHOLENAME added to the CUSTOMER Master File that calls the DEFINE FUNCTION:

```
DEFINE WHOLENAME/A40 = DF.DMFUNCS.DMPROPER(LASTNAME, FIRSTNAME);
     DESCRIPTION = 'Calls DMPROPER to create full name',$
```

The following request uses the DEFINE field WHOLENAME:

```
TABLE FILE CUSTOMER
PRINT WHOLENAME AS Whole,Name
BY ID_CUSTOMER
WHERE ID_CUSTOMER LT 600
ON TABLE SET PAGE NOPAGE
END
```

The output is:

```
             Whole
ID Customer  Name
-----------  -----
         15  Nolan, Tyler
         20  Bull, Joshua
         78  Wood, Zara
        124  Mckenzie, Callum
        125  Charlton, Bradley
        132  Griffiths, Henry
        152  Rowe, Anthony
        161  Storey, Max
        185  Thomas, Evie
        201  Birch, Brandon
        213  Parry, Maisie
        239  Barrett, Taylor
        258  Lord, Harvey
        270  Bell, Jay
        312  Dunn, Daisy
        352  Mckenzie, Callum
        379  Fisher, Leo
        454  Day, Zak
        472  Howarth, Molly
        503  Barrett, Daniel
        531  Hargreaves, Chloe
        566  Fitzgerald, Bethany
```

Using Date System Amper Variables in Master File DEFINEs

Master File DEFINE fields can use Dialogue Manager system date variables to capture the system date each time the Master File is parsed for use in a request.

The format of the returned value for each date variable is the format indicated in the variable name. For example, &DATEYYMD returns a date value with format YYMD. The exceptions are &DATE and &TOD, which return alphanumeric values and must be assigned to a field with an alphanumeric format. The variable names &DATE and &TOD must also be enclosed in single quotation marks in the DEFINE expression.

The variables supported for use in Master File DEFINEs are:

- &DATE

- &TOD
- &DATEMDY
- &DATEDMY
- &DATEYMD
- &DATEMDYY
- &DATEDMYY
- &DATEYYMD
- &DMY
- &YMD
- &MDY
- &YYMD
- &MDYY
- &DMYY

Note that all other reserved amper variables are not supported in Master Files.

Example: **Using the Date Variable &DATE in a Master File DEFINE**

The following version of the EMPLOYEE Master File has the DEFINE field named TDATE added to it. TDATE has format A12 and retrieves the value of &DATE, which returns an alphanumeric value and must be enclosed in single quotation marks:

```
FILENAME=EMPLOYEE, SUFFIX=FOC
SEGNAME=EMPINFO,  SEGTYPE=S1
 FIELDNAME=EMP_ID,       ALIAS=EID,     FORMAT=A9,       $
 FIELDNAME=LAST_NAME,    ALIAS=LN,      FORMAT=A15,      $
 FIELDNAME=FIRST_NAME,   ALIAS=FN,      FORMAT=A10,      $
 FIELDNAME=HIRE_DATE,    ALIAS=HDT,     FORMAT=I6YMD,    $
 FIELDNAME=DEPARTMENT,   ALIAS=DPT,     FORMAT=A10,      $
 FIELDNAME=CURR_SAL,     ALIAS=CSAL,    FORMAT=D12.2M,   $
 FIELDNAME=CURR_JOBCODE, ALIAS=CJC,     FORMAT=A3,       $
 FIELDNAME=ED_HRS,       ALIAS=OJT,     FORMAT=F6.2,     $
DEFINE TDATE/A12   ='&DATE';, $
   .
   .
   .
```

The following request displays the value of TDATE:

```
TABLE FILE EMPLOYEE
PRINT LAST_NAME FIRST_NAME HIRE_DATE TDATE AS 'TODAY''S,DATE'
WHERE LAST_NAME EQ 'BANNING'
END
```

The output is:

LAST_NAME	FIRST_NAME	HIRE_DATE	TODAY'S DATE
BANNING	JOHN	82/08/01	06/17/04

Example: **Using the Date Variable &YYMD in a Master File DEFINE**

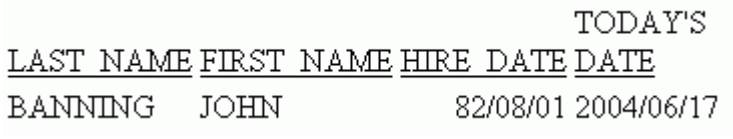
The following version of the EMPLOYEE Master File has the DEFINE field named TDATE added to it. TDATE has format YYMD and retrieves the value of &YYMD:

```
FILENAME=EMPLOYEE, SUFFIX=FOC
SEGNAME=EMPINFO,  SEGTYPE=S1
 FIELDNAME=EMP_ID,        ALIAS=EID,     FORMAT=A9,       $
 FIELDNAME=LAST_NAME,     ALIAS=LN,      FORMAT=A15,      $
 FIELDNAME=FIRST_NAME,    ALIAS=FN,      FORMAT=A10,      $
 FIELDNAME=HIRE_DATE,     ALIAS=HDT,     FORMAT=I6YMD,    $
 FIELDNAME=DEPARTMENT,    ALIAS=DPT,     FORMAT=A10,      $
 FIELDNAME=CURR_SAL,      ALIAS=CSAL,    FORMAT=D12.2M,   $
 FIELDNAME=CURR_JOBCODE, ALIAS=CJC,      FORMAT=A3,       $
 FIELDNAME=ED_HRS,        ALIAS=OJT,     FORMAT=F6.2,     $
DEFINE TDATE/YYMD   = &YYMD ;, $
   .
   .
   .
```

The following request displays the value of TDATE:

```
TABLE FILE EMPLOYEE
PRINT LAST_NAME FIRST_NAME HIRE_DATE TDATE AS 'TODAY''S,DATE'
WHERE LAST_NAME EQ 'BANNING'
END
```

The output is:



LAST_NAME	FIRST_NAME	HIRE_DATE	TODAY'S DATE
BANNING	JOHN	82/08/01	2004/06/17

Reference: Messages for Date System Amper Variables in Master File DEFINEs

The following message appears if an attempt is made to use an unsupported amper variable in a Master File DEFINE:

```
(FOC104) DEFINE IN MASTER REFERS TO A FIELD OUTSIDE ITS SCOPE: var
```

Parameterizing Master and Access File Values Using Variables

You can define global variables in a Master File and use them to parameterize certain attributes in the Master File and its corresponding Access File. For example, you can parameterize the connection attribute in the Access File with a variable you define in the Master File and then specify the actual connection name at run time.

Syntax: How to Create a Master File Variable

Add variable definitions after the FILE declaration in the Master File:

```
VARIABLE NAME=[&&]var, USAGE=Aln, [DEFAULT=defvalue,][QUOTED={OFF|ON},]$
```

where:

`[&&]var`

Is the name you are assigning to the global variable. When you reference the variable in the Master or Access File, you must prepend the name with two ampersands. However, the ampersands are optional when defining the variable.

`ln`

Is the maximum length for the variable value.

`defvalue`

Is the default value for the variable. If no value is set at run time, this value is used.

`QUOTED = {OFF|ON}`

ON adds single quotation marks around the assigned string for the variable. A single quotation mark within the string is converted to two single quotation marks. OFF is the default value.

Reference: Support for Variables in Master and Access File Attributes

In the Master File, the following attributes can be parameterized with variables: POSITION, OCCURS, REMARKS, DESCRIPTION, TITLE, HELPMESSAGE.

In the DBA section of a Master File, the following attributes can be parameterized: USER, VALUE.

In the Access File, the following attributes can be parameterized with variables: CONNECTION, TABLENAME, START, CHKPT_SAVE, CHKPT_FILE, POLLING, TIMEOUT, MAXLUWS, ACTION, MSGLIMIT, DIRECTORY, NAME, EXTENSION, DATA_ORIGIN, MAXFILES, MAXRECS, OBJECT, PICKUP, TRIGGER, DISCARD, ARCHIVE.

Note: You can concatenate multiple variables to create an attribute value.

Example: Parameterizing Attributes in a Master and Access File

The following request creates an Oracle table named ORAEMP from the FOCUS data source named EMPLOYEE:

```
TABLE FILE EMPLOYEE
SUM LAST_NAME FIRST_NAME CURR_SAL CURR_JOBCODE DEPARTMENT
BY EMP_ID
ON TABLE HOLD AS ORAEMP FORMAT SQLORA
END
```

The following is the Master File created by the request:

```
FILENAME=ORAEMP , SUFFIX=SQLORA , $
  SEGMENT=SEG01, SEGTYPE=S0, $
    FIELDNAME=EMP_ID, ALIAS=EID, USAGE=A9, ACTUAL=A9, $
    FIELDNAME=LAST_NAME, ALIAS=LN, USAGE=A15, ACTUAL=A15, $
    FIELDNAME=FIRST_NAME, ALIAS=FN, USAGE=A10, ACTUAL=A10, $
    FIELDNAME=CURR_SAL, ALIAS=CSAL, USAGE=D12.2M, ACTUAL=D8, $
    FIELDNAME=CURR_JOBCODE, ALIAS=CJC, USAGE=A3, ACTUAL=A3, $
    FIELDNAME=DEPARTMENT, ALIAS=DPT, USAGE=A10, ACTUAL=A10, $
```

The following is the Access File created by the request:

```
SEGNAME=SEG01, TABLENAME=ORAEMP, KEYS=01, WRITE=YES, $
```

Add the following variable definitions to the Master File in order to parameterize the TABLENAME attribute in the Access File and the TITLE attribute for the EMP_ID column in the Master File:

```
FILENAME=ORAEMP, SUFFIX=SQLORA , $
VARIABLE NAME=table, USAGE=A8, DEFAULT=EDUCFILE, $
VARIABLE NAME=emptitle, USAGE=A30, DEFAULT=empid,$
```

Now, in the Master File, add the TITLE attribute to the FIELD declaration for EMP_ID:

```
FIELDNAME=EMP_ID, ALIAS=EID, USAGE=A9, ACTUAL=A9,
      TITLE='&&emptitle', $
```

In the Access File, replace the value for the TABLENAME attribute with the variable name:

```
SEGNAME=SEG01, TABLENAME=&&table, KEYS=01, WRITE=YES, $
```

The following request sets the values of the variables and then issues a TABLE request:

```
-SET &&table = ORAEMP;
-SET &&emptitle = 'Id,number';
TABLE FILE ORAEMP
PRINT EMP_ID LAST_NAME FIRST_NAME DEPARTMENT
END
```

Note that the value for &&emptitle is enclosed in single quotation marks in the –SET command because it contains a special character (the comma). The single quotation marks are not part of the string and do not display on the report output. The column title would display enclosed in single quotation marks if the variable definition contained the attribute QUOTED=ON.

On the report output, the column title for the employee ID column displays the value set for &&emptitle, and the table accessed by the request is the ORAEMP table created as the first step in the example:

```
Id
number     LAST_NAME        FIRST_NAME  DEPARTMENT
------     ---------        ----------  ----------
071382660  STEVENS          ALFRED      PRODUCTION
112847612  SMITH            MARY        MIS
117593129  JONES            DIANE       MIS
119265415  SMITH            RICHARD     PRODUCTION
119329144  BANNING          JOHN        PRODUCTION
123764317  IRVING           JOAN        PRODUCTION
126724188  ROMANS           ANTHONY     PRODUCTION
219984371  MCCOY            JOHN        MIS
326179357  BLACKWOOD        ROSEMARIE   MIS
451123478  MCKNIGHT         ROGER       PRODUCTION
543729165  GREENSPAN        MARY        MIS
818692173  CROSS            BARBARA     MIS
```

Example: Concatenating Variables to Create an Attribute Value

In the following example, the TABLENAME attribute requires a multipart name consisting of a database name, an owner ID, a table prefix, and a static table name with a variable suffix. In this case, you can define separate variables for the different parts and concatenate them.

First, define separate variables for each part:

```
VARIABLE NAME=db,USAGE=A8,DEFAULT=mydb,$
VARIABLE NAME=usr,USAGE=A8,DEFAULT=myusrid,$
VARIABLE NAME=tprf,USAGE=A4,DEFAULT=test_,$
VARIABLE NAME=tsuf,USAGE=YYM,$
```

In the Access File, concatenate the variables to create the TABLENAME attribute. Note that the separator for between each part is a period, but to concatenate a variable name and retain the period, you must use two periods:

```
TABLENAME=&db..&usr..&tprf.table&tsuf,
```

Based on the defaults, the TABLENAME would be:

```
TABLENAME=mydb.myusrid.test_table
```

In a request, set the following values for the separate variables:

```
I-SET &&db=db1;
-SET &&tprf=prod_;
-SET &&tsuf=200801;
```

With these values, the TABLENAME used is the following:

```
TABLENAME=db1.myusrid.prod_table200801
```

Converting Alphanumeric Dates to Dates

In some data sources, date values are stored in alphanumeric format without any particular standard, with any combination of components, such as year, quarter, and month, and with any delimiter. In a sorted report, if such data is sorted alphabetically, the sequence does not make business sense. To ensure adequate sorting, aggregation, and reporting on date fields, can convert the alphanumeric dates into standard date format using a conversion pattern that you can specify in the Master File attribute called DATEPATTERN.

Each element in the pattern is either a constant character which must appear in the actual input or a variable that represents a date component. You must edit the USAGE attribute in the Master File so that it accounts for the date elements in the date pattern. The maximum length of the DATEPATTERN string is 64.

Reference: Usage Notes for DATEPATTERN

- If your original date has elements with no USAGE format equivalent, the converted date will not look like the original data. In that case, if you want to display the original data, you may be able to use an OCCURS segment to redefine the field with the original alphanumeric format and display that field in the request.
- DATEPATTERN requires an ACTUAL format to USAGE format conversion. Therefore, it is not supported for SUFFIX=FOC and SUFFIX=XFOC data sources.

Specifying Variables in a Date Pattern

The valid date components (variables) are year, quarter, month, day, and day of week. In the date pattern, variables are enclosed in square brackets (these brackets are not part of the input or output. Note that if the data contains brackets, you must use an escape character in the date pattern to distinguish the brackets in the data from the brackets used for enclosing variables).

Syntax: How to Specify Years in a Date Pattern

`[YYYY]`

Specifies a four-digit year.

`[YYYY]`

Specifies a four-digit year.

`[YY]`

Specifies a two-digit year.

`[yy]`

Specifies a zero-suppressed two-digit year (for example, 8 for 2008).

`[by]`

Specifies a blank-padded two-digit year.

Syntax: How to Specify Month Numbers in a Date Pattern

`[MM]`

Specifies a two-digit month number.

`[mm]`

Specifies a zero-suppressed month number.

`[bm]`

Specifies a blank-padded month number.

Syntax: How to Specify Month Names in a Date Pattern

`[MON]`

Specifies a three-character month name in upper case.

`[mon]`

Specifies a three-character month name in lower case.

`[Mon]`

Specifies a three-character month name in mixed case.

`[MONTH]`

Specifies a full month name in upper case.

`[month]`

Specifies a full month name in lower case.

`[Month]`

Specifies a full month name in mixed case.

Syntax: How to Specify Days of the Month in a Date Pattern

`[DD]`

Specifies a two-digit day of the month.

`[dd]`

Specifies a zero-suppressed day of the month.

`[bd]`

Specifies a blank-padded day of the month.

Syntax: How to Specify Julian Days in a Date Pattern

`[DDD]`

Specifies a three-digit day of the year.

`[ddd]`

Specifies a zero-suppressed day of the year.

`[bdd]`

Specifies a blank-padded day of the year.

Syntax: How to Specify Day of the Week in a Date Pattern

`[WD]`

Specifies a one-digit day of the week.

`[DAY]`

Specifies a three-character day name in upper case.

`[day]`

Specifies a three-character day name in lower case.

`[Day]`

Specifies a three-character day name in mixed case.

`[WDAY]`

Specifies a full day name in upper case.

`[wday]`

Specifies a full day name in lower case.

`[Wday]`

Specifies a full day name in mixed case.

For the day of the week, the WEEKFIRST setting defines which day is day 1.

Syntax: **How to Specify Quarters in a Date Pattern**

`[Q]`

Specifies a one-digit quarter number (1, 2, 3, or 4).

For a string like Q2 or Q02, use constants before [Q], for example, Q0[Q].

Specifying Constants in a Date Pattern

Between the variables, you can insert any constant values.

If you want to insert a character that would normally be interpreted as part of a variable, use the backslash character as an escape character. For example:

- Use \[to specify a left square bracket constant character.
- Use \\ to specify a backslash constant character.

For a single quotation mark, use two consecutive single quotation marks ('').

Sample Date Patterns

If the date in the data source is of the form CY 2001 Q1, the DATEPATTERN attribute is:

```
DATEPATTERN = 'CY [YYYY] Q[Q]'
```

If the date in the data source is of the form Jan 31, 01, the DATEPATTERN attribute is:

```
DATEPATTERN = '[Mon] [DD], [YY]'
```

If the date in the data source is of the form APR-06, the DATEPATTERN attribute is:

```
DATEPATTERN = '[MON]-[YY]'
```

If the date in the data source is of the form APR - 06, the DATEPATTERN attribute is:

```
DATEPATTERN = '[MON] - [YY]'
```

If the date in the data source is of the form APR '06, the DATEPATTERN attribute is:

```
DATEPATTERN = '[MON] ''[YY]'
```

If the date in the data source is of the form APR [06], the DATEPATTERN attribute is:

```
DATEPATTERN = '[MON] \[[YY]\]' (or '[MON] \[[YY]]'
```

Note that the right square bracket does not have to be escaped.

Example: Sorting By an Alphanumeric Date

In the following example, is a sequential file containing the following data:

```
June 1, '02
June 2, '02
June 3, '02
June 10, '02
June 11, '02
June 12, '02
June 20, '02
June 21, '02
June 22, '02
June 1, '03
June 2, '03
June 3, '03
June 10, '03
June 11, '03
June 12, '03
June 20, '03
June 21, '03
June 22, '03
June 1, '04
June 2, '04
June 3, '04
June 4, '04
June 10, '04
June 11, '04
June 12, '04
June 20, '04
June 21, '04
June 22, '04
```

In the DATE1 Master File, the DATE1 field has alphanumeric USAGE and ACTUAL formats, each A18:

```
FILENAME=DATE1   , SUFFIX=FIX ,
  DATASET =     , $
  SEGMENT=FILE1, SEGTYPE=S0, $
    FIELDNAME=DATE1, ALIAS=E01, USAGE=A18, ACTUAL=A18, $
```

The following request sorts by the DATE1 FIELD:

```
TABLE FILE DATE1
PRINT DATE1 NOPRINT
BY DATE1
ON TABLE SET PAGE NOPAGE
END
```

The output shows that the alphanumeric dates are sorted alphabetically, not chronologically:

```
DATE1
-----
June 1, '02
June 1, '03
June 1, '04
June 10, '02
June 10, '03
June 10, '04
June 11, '02
June 11, '03
June 11, '04
June 12, '02
June 12, '03
June 12, '04
June 2, '02
June 2, '03
June 2, '04
June 20, '02
June 20, '03
June 20, '04
June 21, '02
June 21, '03
June 21, '04
June 22, '02
June 22, '03
June 22, '04
June 3, '02
June 3, '03
June 3, '04
June 4, '04
```

In order to sort the data correctly, you can add a DATEPATTERN attribute to the Master File that enables to convert the date to a date field. You must also edit the USAGE format to make it a date format. To construct the appropriate pattern, you must account for all of the components in the stored date. The alphanumeric date has the following variables and constants:

- Variable: full month name in mixed case, [Month].
- Constant: blank space.
- Variable: zero-suppressed day of the month number, [dd].
- Constant: comma followed by a blank space followed by an apostrophe (coded as two apostrophes in the pattern).
- Variable: two-digit year, [YY].

The edited Master File follows. Note the addition of the DEFCENT attribute to convert the two-digit year to a four-digit year:

```
FILENAME=DATE1   , SUFFIX=FIX ,
  DATASET =     , $
  SEGMENT=FILE1, SEGTYPE=S0, $
    FIELDNAME=DATE1, ALIAS=E01, USAGE=MtrDYY, ACTUAL=A18,
      DEFCENT=20,
      DATEPATTERN = '[Month] [dd], ''[YY]', $
```

Now, issuing the same request produces the following output. Note that DATE1 has been converted to a date in MtrDYY format (as specified in the USAGE format):

```
DATE1
-----
June  1, 2002
June  2, 2002
June  3, 2002
June 10, 2002
June 11, 2002
June 12, 2002
June 20, 2002
June 21, 2002
June 22, 2002
June  1, 2003
June  2, 2003
June  3, 2003
June 10, 2003
June 11, 2003
June 12, 2003
June 20, 2003
June 21, 2003
June 22, 2003
June  1, 2004
June  2, 2004
June  3, 2004
June  4, 2004
June 10, 2004
June 11, 2004
June 12, 2004
June 20, 2004
June 21, 2004
June 22, 2004
```

Chapter 5

Running Db2 Web Query Reports Using the Java Batch Run Utility

The Java Batch Run utility (RUNWEBQRY) enables you to run a Db2 Web Query report from the command line so that the request can be submitted to a batch queue, without having to physically sign in to Db2 Web Query. The utility accepts a single report procedure (.fex) that is in Db2 Web Query and executes the .fex using a Java program.

In this chapter:

Java Batch Run Utility Prerequisites

The following are prerequisites for running the utility:

- Db2 Web Query must be installed and running on the system where the reporting and application servers are running.
- FOCEXECs which exist in the Db2 Web Query environment.
- An available user ID that is licensed for Db2 Web Query.

Procedure: How to Invoke the Java Batch Run

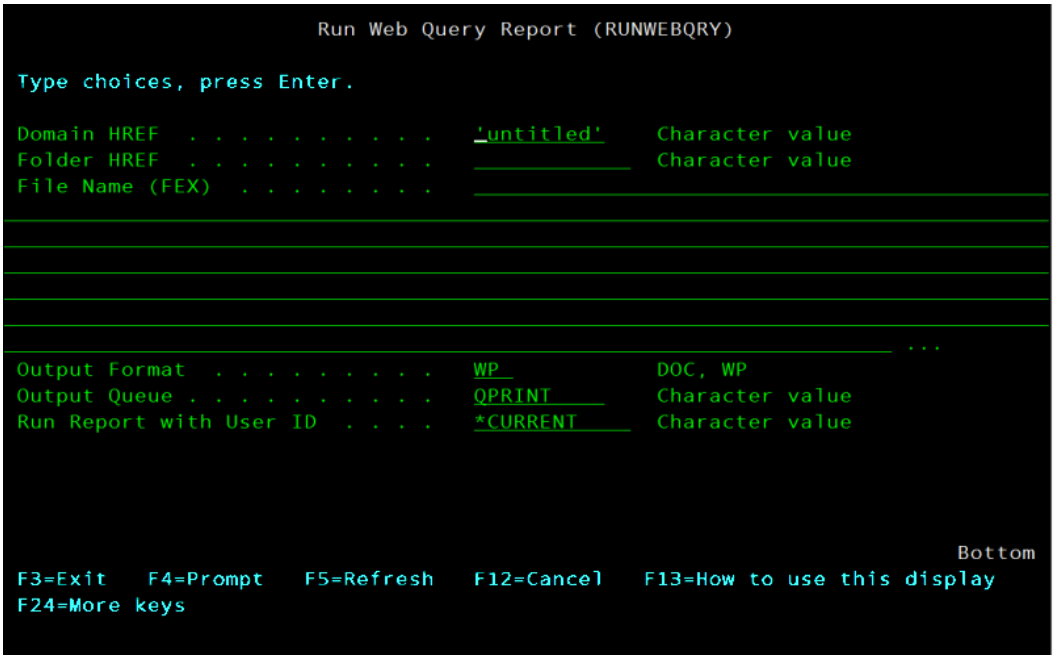
To invoke the utility from the command line:

1. Sign in to the IBM i system using a 5250 terminal emulator.
2. At the command line, execute the following command:

   ```
   RUNWEBQRY
   ```

3. Press the *F4* key.

The Run Java batch in WebQuery (RUNWEBQRY) screen opens, as shown in the following image.



4. On the screen, type values for the following input parameters:

Domain href

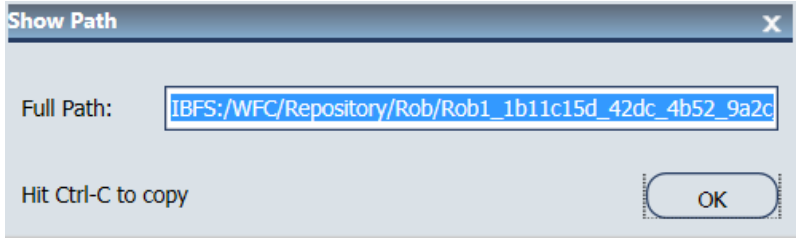
Is the domain name in Dashboard or an href. An href is an internal name given to an object. The href can be displayed using the Properties option in the Dashboard menu, which is available by right-clicking the mouse. You do not need to use the href if the name is exactly 8 characters. You need to use the href if the name is less than 8 characters or more than 8 characters. This input parameter can be retrieved from the Properties page by right-clicking the report name and choosing *Properties* from the drop-down menu.

Folder href

Is the folder name in Dashboard or an href. An href is an internal name given to an object. The href can be displayed using the Properties option in the Dashboard menu, which is available by right-clicking the mouse. You do not need to use the href if the name is exactly 12 characters. You need to use the href if the name is less than 12 characters or more than 12 characters. This input parameter can be retrieved from the Properties page by right-clicking the report name and choosing *Properties* from the drop-down menu.

File Name (fex)

Is the full path or name of the report FOCEXEC (fex) as displayed in the BI portal tree. This input parameter can be retrieved from the Show Path option by right-clicking the report name and choosing *Show Path* from the drop-down menu, as shown in the following image. Press Ctrl+C to copy the full path and report name of the fex.



Output Format

Is the report format. Possible values are DOC or WP. WP is the default value.

Output Queue

Is the name of the outq on the IBM i system where the report will be sent. QPRINT is the default value.

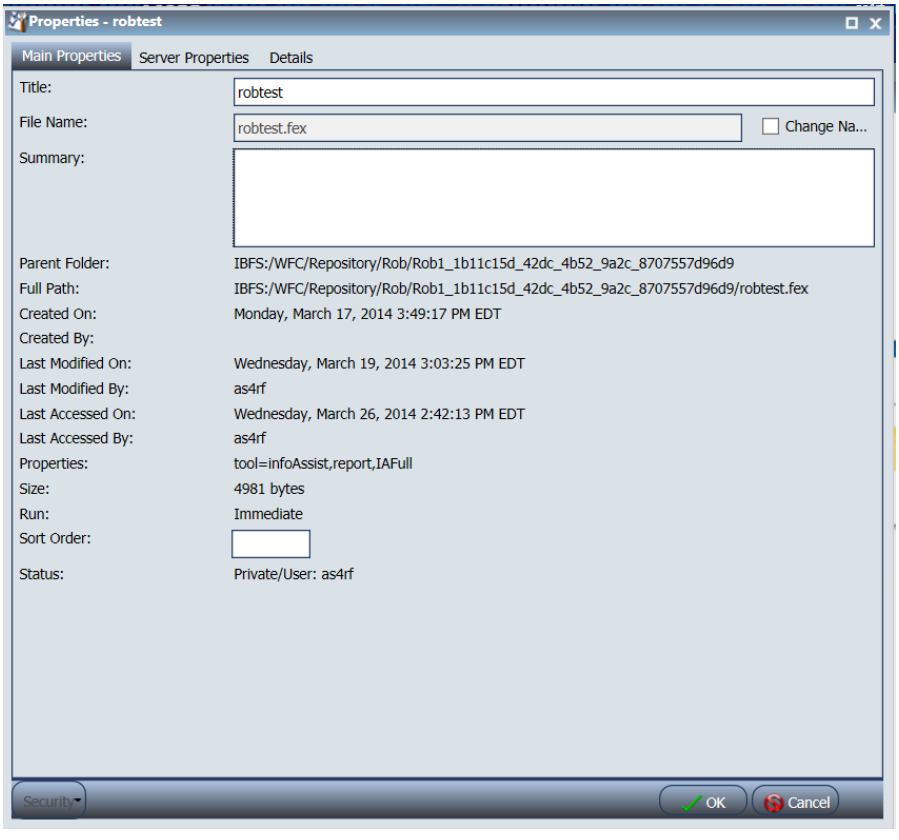
Run Report with User ID

Enables you to submit the job using another user ID. The default value is *CURRENT which means that the current user ID will be used to submit the job.

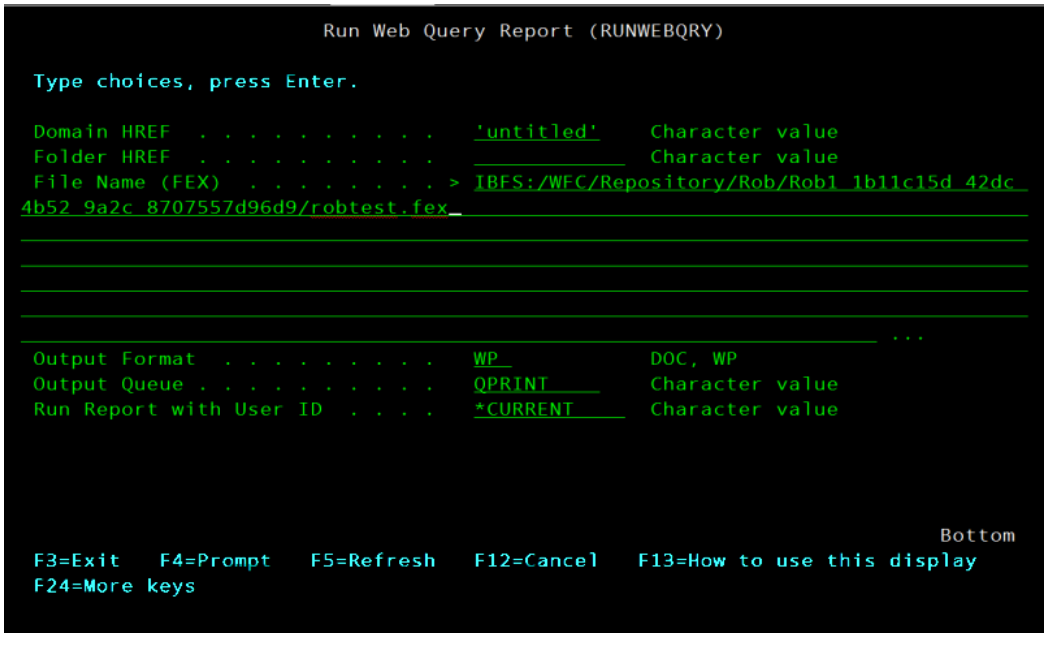
Note: The RUNWEBQRY command can be part of a job stream using SBMJOB to run multiple requests.

Reference: Retrieving Input Parameters for the RUNWEBQRY Command

The following is an example of a Properties page for retrieving parameter information. This page is accessed by right-clicking a report name in a report folder, and choosing *Properties* from the drop-down menu.



The following image shows the input parameters populated from the example properties page.



Chapter 6

Using the Db2 Web Query Viewer

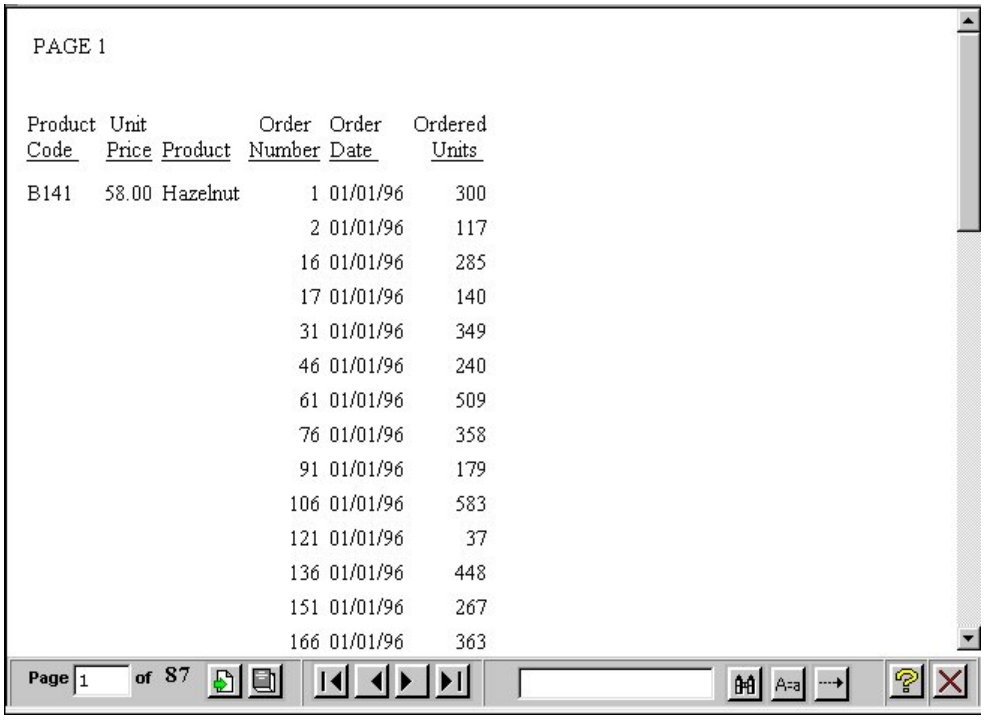
The Db2 Web Query Viewer uses the On-demand Paging facility. When On-demand paging is enabled, Db2 Web Query saves the bulk of your report to your Web server and delivers one page of report output at a time, decreasing the amount of time you wait for your report to process. The bulk of your report remains on the Web server until you request it or close the Viewer.

The Db2 Web Query Viewer improves your ability to handle long reports by allowing you to view a single page of report output.

In this chapter:

Navigating a Report With the Db2 Web Query Viewer

When you run a report designated for On-Demand Paging, the Db2 Web Query Viewer opens automatically and displays the first page of the report. The Viewer consists of two panes: the Report Pane and the Viewer Control Panel, as shown in the following image.



The Report Pane is the larger pane and contains one page of report output. When you first run a report, the Report Pane contains the first page of report output. The Viewer Control Panel contains the controls that allow you to display specific pages, deliver the entire report to your web server, and search your document for particular strings of information.

Syntax: How to Enable the Db2 Web Query Viewer

```
SET WEBVIEWER = {OFF|ON}
```

or

```
ON TABLE SET WEBVIEWER {OFF|ON}
```

where:

OFF

Disables on-demand paging. Db2 Web Query downloads the entire report to a standard browser window. OFF is the default value.

ON

Enables on-demand paging. Db2 Web Query downloads the first page of a report to the browser in the Viewer. The number of lines displayed at one time depends on Windows desktop settings (resolution).

Example: **Enabling the Db2 Web Query Viewer**

The procedure and launch page for this example are run in Db2 Web Query. They must be tested and run in this environment.

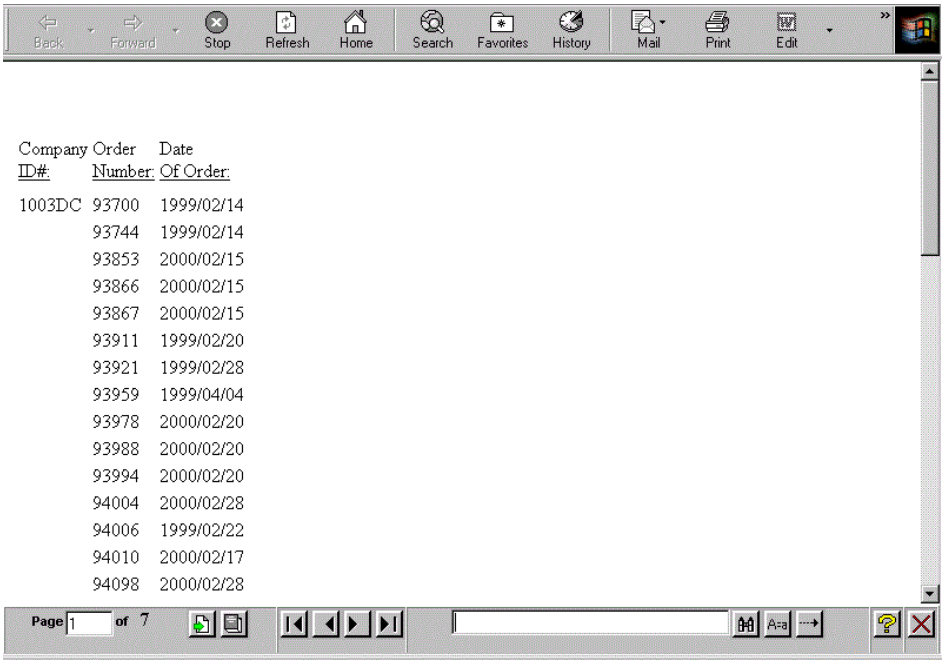
1. Create a procedure named ONDEMAND, which displays an order report for a store in the Viewer.

   Procedure:

   ```
   SET WEBVIEWER=ON
   TABLE FILE CENTORD
   PRINT ORDER_NUM ORDER_DATE
   BY STORE_CODE
   WHERE STORE_CODE EQ '1003DC'
   ON TABLE SET PAGE-NUM OFF
   ON TABLE SET STYLE *
   TYPE=REPORT, GRID=OFF,$
   ENDSTYLE
   END
   ```

2. Create a launch page from which a user can run the report.

3. Run the launch page, and click the link. The report displays in the Viewer.



Opening and Closing the Db2 Web Query Viewer

You can specify a target frame in which to open the Db2 Web Query Viewer, and a home page that displays when you close the Viewer.

Syntax: How to Open the Db2 Web Query Viewer in a Target Frame

```
SET WEBVIEWTARG = {target_frame|OFF}
```

where:

target_frame

Is the name of an existing frame in the browser or one of the following reserved HTML target frames:

`_blank`

Opens the Viewer in a new browser window. This is the default for reports that do not have accessibility enabled.

`_self`

Opens the Viewer in the same frame as the anchor.

`_parent`

Opens the Viewer in the immediate parent frame that contains the anchor.

`_top`

Opens the Viewer in the current browser window.

`OFF`

Opens the Viewer in the frame from which you ran the report. This is the default for reports that have accessibility enabled.

Syntax: **How to Display a Home Page When You Close the Db2 Web Query Viewer**

```
SET WEBVIEWHOME = {home_URL|OFF}
```

where:

home_URL

Is a valid URL that displays an HTML page when you close the Viewer.

OFF

Displays a blank browser window when you close the Viewer. You must enter another URL to run another report. OFF is the default value.

Reference: **Closing the Db2 Web Query Viewer**

The Close button, located on the Control Frame, closes the Viewer and removes the report from the web server. The page the browser displays next depends on the WEBVIEWTARG and the WEBVIEWHOME settings, as follows:

- If you set WEBVIEWTARG to _blank, the window that contained the Viewer is removed. The browser does not display any page in any frame, and the WEBVIEWHOME setting has no effect.
- If you set WEBVIEWTARG to any other value, the result of clicking *Close* depends on the WEBVIEWHOME setting:
  - If you set WEBVIEWHOME to a URL, the browser displays the page associated with the URL in the frame that the Viewer occupied.
  - If you set WEBVIEWHOME to OFF, the browser displays a blank page.

Controlling Button Display on the Db2 Web Query Viewer

You can issue commands that specify whether the Viewer displays the Close, Help, and All Pages buttons.

Syntax: How to Control Whether The Close Button Displays

```
SET WEBVIEWCLOSE = {OFF|ON}
```

or

```
ON TABLE SET WEBVIEWCLOSE {OFF|ON}
```

where:

ON

Displays the Close button. ON is the default value.

OFF

Does not display the Close button.

Syntax: How to Control Whether The Help Button Displays

```
SET WEBVIEWHELP = {OFF|ON}
```

or

```
ON TABLE SET WEBVIEWHELP {OFF|ON}
```

where:

ON

Displays the Help button. ON is the default value.

OFF

Does not display the Help button.

Syntax: How to Control Whether The All Pages Button Displays

```
SET WEBVIEWALLPG = {OFF|ON}
```

or

```
ON TABLE SET WEBVIEWALLPG {OFF|ON}
```

where:

ON

Displays the All Pages button. ON is the default value.

OFF

Does not display the All Pages button.

Procedure: How to Enable the Goto Last Page Button

1. From Internet Explorer, click the *Tools* menu and select *Internet Options*.
2. The Internet Options dialog box opens.
3. From the Temporary Internet files box, click *Settings*.

   The Settings dialog box opens.
4. Click the *Every visit to the page* radio button.
5. Click *OK* to apply the change and exit the Settings dialog box.
6. Click *OK* to exit the Internet Options dialog box.

Using the Viewer Control Panel

The Viewer Control Panel, as shown in the following image, (located at the bottom of the window) contains the controls you use to navigate through the report and to search for a string in the report. The Viewer Control Panel navigational controls allow you to display the next or previous page, the first or last page, or a specific page. You use the searching function to have the Viewer locate a search string you specify within all report pages.

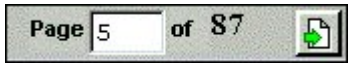
Note: When specifying a search string, you must specify the actual number of spaces between characters because HTML displays a single space, even when multiple spaces are used between characters.

Procedure: How to Navigate Through a Report

The Viewer Control Panel offers several ways to view pages in your report:

❑ To display a specific page:

1. Enter a page number in the Page input box, as shown in the following image.



2. Click Go to Page, as shown in the following image.

- To display the previous or the next page in sequence, click *Previous* or *Next*, as shown in the following image.

- To display the first or last page of the report, click *First Page* or *Last Page*, as shown in the following image.

- To download the entire report to the browser as a single document, click *All Pages*, as shown in the following image.

- To close the Viewer, click *Close*, as shown in the following image.

Searching a Report

The Viewer Control Panel contains controls that offer several ways to search your report. Using the Viewer search controls, you can select a string of information, such as a phrase that occurs in your report or a group of numbers, and search for each occurrence of that string. You can further customize your search by matching capitalization of words exactly (a case-sensitive search) or by controlling the direction of your search (either forward or backward from your starting point in the report). Use these controls to search your report:

- To perform a case-sensitive search, click *Match Case*, as shown in the following image.

- To search backward in a report, click *Search Backward*, as shown in the following image.

- To locate a specific string, type the string you want to search for and click *Find*, as shown in the following image.

Procedure: How to Search the Report

1. Enter the string in the Search input box.
2. Click *Match Case* if you want to perform a case-sensitive search.

   Notice that the Viewer displays the Match Case button with a red line across it to indicate that it is active.
3. To begin your search, click:

   - *Search Backward* to search for the string from the current page back to the first page.

     Or

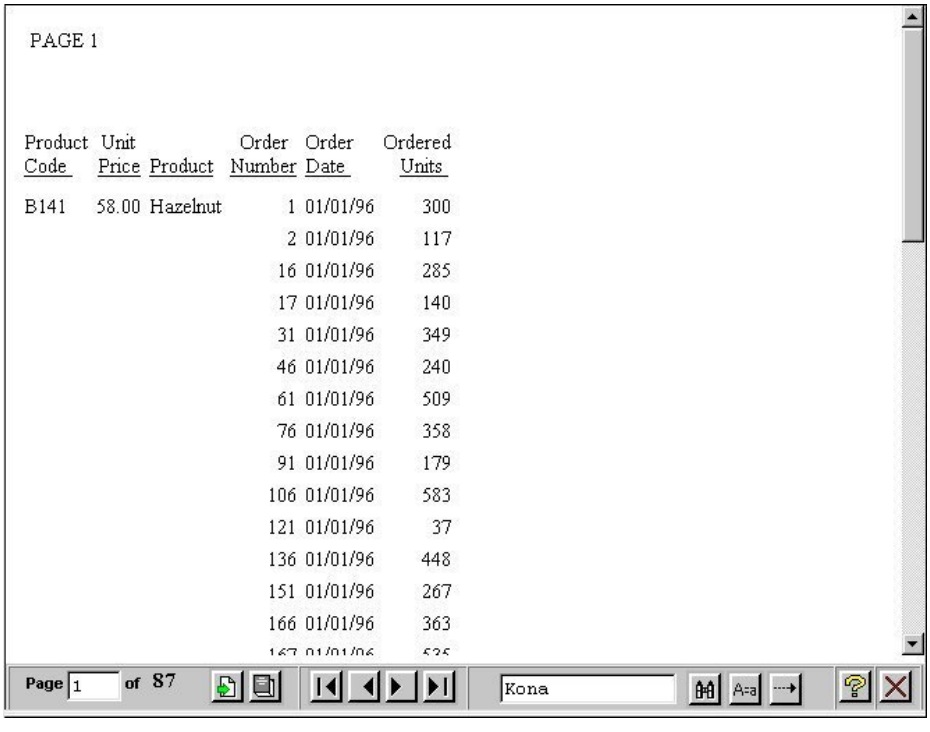
   - *Find* to search from the current page to the end of your report.

   The Viewer searches the report, underlines the first occurrence of the string, and opens the display to the top of the page on which it appears. If the underlined occurrence of the string is not visible, scroll down the page until it appears in the window of the Viewer.
4. Click *Find* again to search for another occurrence of the string.

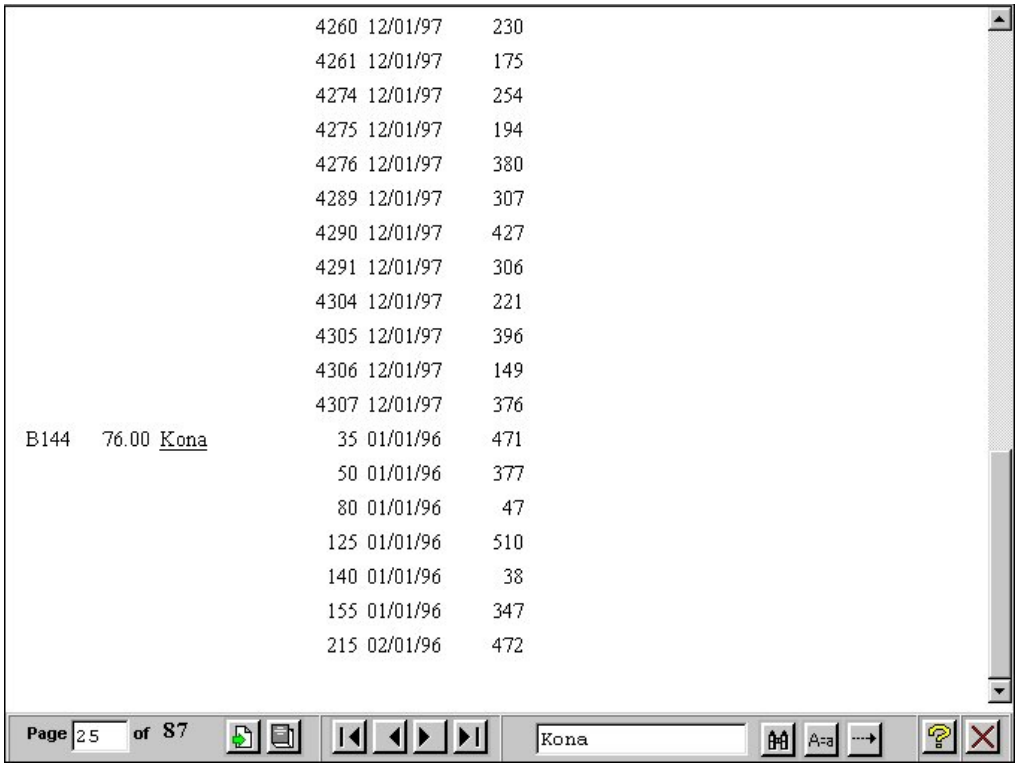
Example: Using the Viewer Control Panel to Search

You want to use the Viewer Control Panel to navigate a long report called Coffee Sales to find occurrences of the string "Kona," a type of coffee that you sell. After you run the report, Db2 Web Query displays the first page of the report in the Viewer.

1. To search for sales of Kona, type Kona in the input box and click *Find*, as shown in the following image.



The Viewer returns your report with the first occurrence of your search string underlined, as shown in the following image.



2. Click *Find* again to locate the next occurrence of Kona.

Index

I

J

L

M

N

O

P

Q

R

S

T

U

V

W

Y