InfoBurst User Guide

Version 2.5
## Contents

### Introduction
- Documents and Platforms supported ................................................................. 6
- The ‘Burst’ .......................................................................................................................... 7
- Summary of Key Features .............................................................................................. 8
- Example of the ‘Burst’ approach .................................................................................. 10
- Operational Control ...................................................................................................... 11

### Getting Started with InfoBurst
- Logon ............................................................................................................................... 12
- System security and Folder structure .......................................................................... 13
- Navigating the user interface ....................................................................................... 14
- Common screen icons and buttons ............................................................................ 15
- Adding the first documents to the Catalog ................................................................ 16
- Create, Execute and Monitor a simple Burst ............................................................... 17
- QuickBurst ..................................................................................................................... 21
- Introduction to Report Bursting .................................................................................. 22

### Burst Management in detail
- Adding or removing documents ............................................................................... 24
- Changing the document name ................................................................................. 24
- Setting document specific options ......................................................................... 25
- Conditional Delivery using $IB$DELIVER variable ................................................. 28
  - Using $IB$variables in document to allow conditions to vary ......................... 31
- Supplying a document Parameter .......................................................................... 35
  - Multi-value parameters ......................................................................................... 35
  - InfoBurst parameter Lists (IPL and EPL) to use report bursting .................. 36
- Creating a new distribution ...................................................................................... 37
- Using a distribution template ................................................................................. 37
- Applying a distribution template to multiple documents in the burst ............... 37
- Global email options ............................................................................................... 38
  - Email compression ................................................................................................. 40

### Distribution options in detail
- Destinations in detail ............................................................................................... 41
  - Network Share ........................................................................................................ 42
  - Email ...................................................................................................................... 44
  - FTP ....................................................................................................................... 45
  - InfoBurst Server .................................................................................................... 47
  - WebDAV ............................................................................................................... 48
  - BusinessObjects Repository ................................................................................. 50
  - XI Platform .......................................................................................................... 52
  - Printer .................................................................................................................. 53
- Formats in detail ........................................................................................................ 54
  - Native .................................................................................................................... 54
  - PDF ....................................................................................................................... 54
Introduction

InfoBurst is a powerful and flexible report distribution and publishing solution that delivers business critical information to the right people, in the right format at the right time.

InfoBurst supports a wide variety of document types and platforms and offers unrivaled functionality and flexibility to be able to handle everything from a simple report distribution task right up to a complex multi-document report bursting scenario.

Since its introduction in 2002, InfoBurst has evolved into the most capable product of its kind and provides a clean and consistent approach to report distribution and bursting while hiding the complexities of the underlying document type or platform.

For example, if you currently use BusinessObjects 5.x or 6.x and are planning to migrate to an XIR2 platform you will find that the way that InfoBurst handles your report distribution challenges will be the same, in that the workflow and the way that ‘bursts’ are defined and executed are consistent even though the 5.x/6.x and XIR2 environments are totally different.

In addition to supporting the traditional Business Intelligence documents such as BusinessObjects FullClient, Web Intelligence, Crystal Reports and Microsoft Reporting Services, InfoBurst also has full support for Crystal Xcelsius dashboards and can provide innovative solutions for both distributing these dashboards and also managing and providing dynamic data that can be extracted from either a BI report or even the underlying database. For full details on these exciting features, please refer to the section “Innovative solutions for Crystal Xcelsius”.

InfoSol™
**Documents and Platforms supported**

What sets InfoBurst apart from other solutions is that it supports a wide variety of different documents and platforms in a single, consistent interface that offers the ultimate in terms of power, flexibility and control.

InfoBurst supports the following document types and platforms:

- BusinessObjects 5.x Full Client
- BusinessObjects 6.x Full Client
- BusinessObjects Desktop Intelligence Full Client
- BusinessObjects WebI 6.x
- Crystal Reports 9, 10 and XI Client
- Crystal Reports XI (XIR1 Platform)
- Crystal Reports XI (XIR2 Platform)
- BusinessObjects WebI (XIR1 Platform)
- BusinessObjects WebI (XIR2 Platform)
- BusinessObjects Desktop Intelligence (XIR2 Platform)
- Microsoft Reporting Services v1
- Crystal Xcelsius (.swf)

The following “static” document formats are also supported:

- PDF
- Word
- Excel
- Text
- PowerPoint
- SWF (Flash or Swish)
- Zip
The ‘Burst’

InfoBurst is unique in its approach to report distribution; it uses a concept called a “Burst” to group together documents of any type rather than just work on a single document. This greatly reduces the number of “objects” that need to be created and managed and is very useful to consolidate information. For example, email recipients within a burst will receive only one email, but depending on the distribution conditions they could receive multiple file attachments or notifications that include a direct link to the report.

Each document in a burst can have multiple delivery options, each of which offers a comprehensive choice of format and destination.

The available destinations are:

- Network Share
- Email Attachment
- Email in-line (embedded content with images)
- FTP
- Secure FTP
- WebDAV share (web folders)
- InfoBurst Server
- BusinessObjects repository (5.x and 6.x)
- XIR1 platform (instance or new object in a folder)
- XIR2 platform (new object in a folder)

The available formats are:

- Native
- PDF
- Excel
- HTML
- Text (Plain, CSV)
- XML
Summary of Key Features

There are so many features and functions in the product that are not obvious at first glance so here is a list of some of the most used and more important features that you should be aware of:

- Powerful report bursting options (single and multi-pass or both)
- Mix and match any document type (BusObj,Crystal,Microsoft) in a burst
- Enterprise Document Providers (EDP) allow for multiple connections for a single type, so you could define both a test and production XIR2 platform
- Templates can be defined for any distribution to avoid re-tying each time
- Date prompts for a document can be supplied using macros with any format string and date calculations are supported (if you need yesterday or last day of month)
- Multi-pass bursting can be easily setup for any document by using an IPL or EPL to ‘drive’ the refresh (IPL=Internal Parameter List, EPL=External Parameter List)
- Email distribution to a dynamic list of recipients can be done using an EDL (External Distribution List) which can be a file or a dynamic query
- EPL/EDL can be a dynamic database query against virtually any database
- Email using HTML format converts the report and any graphics to “in-line” format meaning no attachments
- You can configure a backup Email server and InfoBurst will switch to this if it fails to connect to the primary server
- Zip compression can be used for Email attachments either each file or all files delivered as part of the burst
- Zip compression can use a password and optional data encryption to secure the contents
- When distributing to a network share, FTP site or WebDav share, InfoBurst will automatically create any sub-folders as needed
- Distribution macros can be used to fully control the directory and/or file-name generated (e.g. store file on \rptserver\Daily\=[State]\Sales for [=City].pdf will take the current values for State and City and create a new sub-folder for each state and create a PDF with a name that includes the city).
- You can run your own script using Network distribution if you wish to perform a specific action such as onward publish to your own portal system (InfoBurst will deliver the file to the network and invoke your command-line)
- Report-tab selection is available for BusObj/WebI documents allowing you to only deliver certain tabs (you can even send one tab as PDF and the other in Excel if you wanted to).
- Email notification allows you to publish a file to a central location such as a network share and send an email with a link to the file
- When scheduling a burst (or multiple bursts) you can use Events (one or more) to wait for an external dependency to be true (such as when the datamart has loaded). Events can be database queries, files or Boolean triggers.
- If you want to be informed if a burst aborts, you can setup automatic burst abort emails for all bursts or just selected high-priority bursts
• InfoBurst can deliver data to an Xcelsius dashboard using either a BI document or a series of database queries as the data source. The data can be arranged into various formats such as a crosstab and the resulting XML data is cached on the InfoBurst server for quick access.
• InfoBurst can distribute Xcelsius dashboards using email and embed the dashboard into the message.
• InfoBurst can create an off-line ‘Dashboard Package’ for dashboards that use dynamic XML data to allow for off-line viewing (you do not need access to the server).
• When bursting a Desktop Intelligence report, InfoBurst will use both the XIR2 server and also the local InfoBurst server to maximize the throughput.
• You can create your own macros using System->UserFunctions if you need to re-use a common macro, for example you could create a macro called YESTERDAY that returns the date for yesterday in a certain format. You can use user functions everywhere you can use macros (prompt values, distribution etc).
• If you want to customize an Excel output, you can catalog a workbook with InfoBurst and use it as a base and optionally run a macro to perform any specific tasks before the file is distributed.
• When performing a bursting operation using email distribution, any EDL (External Distribution List) query will automatically be supplied with the current bursting values, so you if you were bursting by State (AZ,CA..) you could easily retrieve all sales managers email addresses where State=? and InfoBurst will supply AZ for the first cycle then CA etc.
• FTP can be normal, secure using SFTP or secure using S/Shell options.
• You can deliver documents back to the InfoBurst server and control the number of versions to keep. Users can view the documents by logging into InfoBurst (this is a good option if you don’t have InfoView).
• It is possible to ignore a distribution if there was no data after a refresh.
• For BusinessObjects FullClient documents you can select which data providers to refresh.
• For BusinessObjects FullClient documents it is possible to extract data elements while bursting (e.g. burst on customer, store by zipcode).
• Using the AtEnd option for a schedule you can set a trigger that another schedule is waiting for (this allows for ScheduleA to run first, then run ScheduleB).
• System overview and Document usage dashboards are available as an option to show you the system burst/document throughput and most used documents.
**Example of the ‘Burst’ approach**

A burst can be as simple as a single document with a single distribution or it can be as complex as you like. There are no restrictions on how many documents you can add to a single burst, or any restriction on the mix of document types.

If you have a BI deployment that consists of a mixture of document types (e.g. BusinessObjects FullClient/DeskI and Web Intelligence) or Crystal and Microsoft Reporting Services you can freely add any combination that you like to a burst.

A burst can be changed at any time; you can add or remove documents, edit existing distributions or add new ones as needed. Documents within a burst can be disabled if you need to skip them at any time and individual distributions can also be disabled. These features help when testing and can be useful in production also if you need to re-run a burst in the event of a failure.

When you create a burst, you will most often be driven by what you are trying to achieve and these ‘real world’ scenarios are what InfoBurst does best.

To help get the basic idea of a Burst, let’s consider this simple requirement:

A BusinessObjects full-client document with a date prompt needs to be refreshed and delivered to a number of external Sales Managers via email, and a group of internal financial analysts also need the information but they prefer Excel format.

With InfoBurst, we can create a burst and select the document from the catalog and for the parameter we can use some InfoBurst date macros to fill in the date in the correct format (for example, if we wanted yesterdays date in YYYY-MM-DD format we could use the syntax `[TODAY-1,YYYY-MM-DD]`).

For the distribution, you can create one to handle the email, using either PDF format if they prefer an attachment or HTML if they just want to view the report as easily as possible (and this can render on some capable PDA devices also) and a second distribution for the same document using Excel format and publish to an internal network share (if you wanted, you could add an email notification to inform them the report is available and provide a link to the file).

As your requirements change, you could easily edit the burst and add a second or third document and add the distributions as needed. When using email distribution, InfoBurst will always group together attachments and notifications per recipient so that where possible one email is sent to each person rather than a flood of emails each with a single message or attachment. By default InfoBurst will allow up to 5mb of attachments per message and this value can be changed to suit your email server limits.
**Operational Control**

InfoBurst excels at ‘Enterprise’ report distribution, meaning that reports are scheduled to be delivered to a wide variety of recipients and destinations, typically overnight and often these schedules are coordinated with external events (such as when the datamart has been loaded). For large and complex reports, true report bursting is often used to break a report by some key data element (such as customer or state).

As you requirements grow, so does the number of reports that you will process, and the number of people that will rely on getting their reports in a timely fashion. If there is a failure to deliver this information for any reason (database error, network issue etc) it is critical for the administrators of InfoBurst to be able to troubleshoot, re-run bursts and find out what has happened.

InfoBurst has very complete operational auditing and logging so that you always know what is scheduled, what is running and what has previously run.

There are three main logs that can be viewed on-line:

1. Burst activity logs
2. System usage audit logs
3. Document delivery logs

For every burst that runs, an Activity Log is created that records a detailed step by step account of what the burst did, for example connect to BusinessObjects, open a document, supply a parameter, refresh the data, save in PDF.

Each step has a status and the burst as a whole has a status, so in the event of a failure it is possible to filter the display to show only those bursts that have aborted and then to see each step of the burst.

Any aborts can be automatically emailed to a support group who will receive a copy of the detailed activity log.

As users log-on and perform tasks such as creating or editing bursts and schedules, the Audit Log records each action so that an Administrator can review who did what and when.

For each separate delivery (Network share, FTP, Email etc) InfoBurst creates an entry in the Delivery Log and records information such as date, time, size of file, format and location. The administrator can search the delivery logs by either document or by location to verify that the report was delivered (for example, a search by email recipient on a particular date can confirm that a report was emailed).
Getting Started with InfoBurst

This chapter will show you the basics, how to logon on to the system, navigate the user interface and how to perform a basic workflow where you will work with documents and create your first distribution.

**Logon**

InfoBurst is a secure application and you must enter a valid username and password to gain access.

Open your browser and enter the URL that corresponds to your InfoBurst website. As an example, if your web server hosting InfoBurst is called ISWEB, navigate to http://ISWEB/InfoBurst.

You should see a screen like this:

![InfoBurst Logon Screen](image)

Enter a valid user name and password. The system is initially configured with a user called ADMIN with the same password. The password should be changed and at least one other admin user should be created. Use the Admin->User Management option to do this.
**System security and Folder structure**

InfoBurst users are either Administrators or not and as an Admin user all functions are available and all “objects” such as documents, bursts and schedules are displayed.

Non-Admin users have a more limited feature set and their menu will not feature certain functions. Non-Admin users can be given certain rights that will allow them to manage their own report distribution tasks if you wish.

Users can be added to Groups if required to allow for easier management.

InfoBurst uses a folder type system to allow you to organize documents, bursts and schedules and access can be given to non-Admin users if required. Access can be given at both Group and User level.

For example, an Admin user could create a new top-level folder called “Sales Reports” where certain documents are stored. A user could be added to a group called “SalesDept” and this group could be given access to the “Sales Reports” folder.

All users have a “home” or “root” folder where they can store documents, bursts and schedules. This folder will be displayed as ”My Documents” or “My Bursts”.

Using the **Documents->Manage FileSystem** menu, you can create new folders and also move objects between folders.
Navigating the user interface

At the top of each screen is a header and menu.

Note: if you see a red symbol in the top right corner, it means that the InfoBurst back-end service is not running and some functions will not be available (such as cataloging a new document or adding/editing a schedule).

The menu is organized around the following major groups:

Documents

- Manage documents that are known to InfoBurst
- Add new documents from the network or a BI platform such as WebI or XI
- Manage the folders on the system and create new ones, set access rights on folders
- Move documents (or other objects) to a different folder
- Change the ownership of certain objects

Bursts

- Manage bursts or create new bursts.

Schedules

- Create or Edit re-occurring schedules
- Manage external events used by schedules
- View the schedule for today or a future date.
- Create or Edit Calendars

Admin

- Manage users and groups
- Create or Edit Distribution lists and templates
- Create or Edit Parameter lists for use with report bursting
- Manage Database connections used by parameter/distribution lists
- Manage Xml Data Sources for Xcelsius dashboards
System

- View or Edit the system configuration
- Manage Enterprise Document Providers such as WebI and XI
- Perform housekeeping and system maintenance tasks
- View the server log
- Create or Edit user functions
- Manage the list of bursts on the alert list

Logs

- View the activity log for a burst or search the activity logs
- View the delivery logs for a document or location
- View the system audit log to see who did what and when

General

- View system dashboards (if available)
- Log-off

Help

- Display license information
- View the What’s new in this release document
- View the version history notes

Common screen icons and buttons

When working with documents, bursts, schedules etc, you will typically see action buttons such as [New Burst] and [Delete] that clearly show their purpose.

InfoBurst uses the [Edit] symbol to indicate that the item can be edited therefore clicking on this icon will always take you to an edit screen.

If you are not sure what an icon does, hover the mouse over the icon for a few seconds to see a brief description.
Adding the first documents to the Catalog

If you are starting with an empty system, your first task is to add some documents to InfoBurst, this is a process known as “cataloging”. InfoBurst will open the document to discover information such as any prompts/parameters that are required.

As InfoBurst supports a number of document types and platforms, the first thing to check is that the system is configured to reflect the correct types and versions.

Use the System->Configuration menu and review the settings under the INFOBURST category for BOVersion and CRVersion to ensure they are correct.

If you wish to catalog a document from a WebI or XI platform, use the System->Enterprise Document Providers menu to create a connection to the platform.

For BusinessObjects documents (and Crystal Reports on XIR1/XIR2) you should also ensure that your platform logon credentials have been setup in your user profile. Use the Admin->User Management menu and check the settings for the ”Provider Logon” tab.

To catalog a client-document such as BusinessObjects or Crystal Reports use the Documents->Add New Document menu and browse for the file. You can choose the name and folder location for the new document and optionally “link” to this report so that InfoBurst will always take the latest copy from the network each time the document is used.

To catalog a document from a configured BI platform such as XIR2, use the Documents->Manage Enterprise Documents menu or if there is only one platform configured, use the Documents->Manage documents on XX (where XX is the server) option. Documents will be stored under the configured EDP root folder with a matching folder to that of the BI platform. For example, if you catalog a WebI document from the “Sales” folder, InfoBurst will store it in the root folder:Sales folder with the same name.

Once you have a document cataloged, you can use the Documents->Catalog Management menu to see the folder structure and view more detailed document information.
Create, Execute and Monitor a simple Burst

Now that documents exist in the catalog, you are free to create a burst to refresh and deliver the report. There are a couple of ways of getting to the new burst screen, if you are already on the catalog management screen you can select the document and press the “New Burst” button or you can start a new burst from scratch using Bursts->Create New Burst.

Give the burst a name and an optional description and select the document(s) to process (if you came from catalog management you should see your selected document already).

By default, the burst will be created in your “My Bursts” folder but you can change this if you wish.

Press “Continue” to see the Burst Details screen. This screen shows you the documents in your burst and allows you to define parameters and distributions.

The Document Instance column shows each document’s path in the InfoBurst folder structure and also its name as it will be used in the burst. You can easily change this name (and set other attributes such as Refresh) by clicking on the blue name link.

In this example, the document selected does require some prompts to be supplied so we will click on the parameter icon to set these up.
There are a number of ways of supplying values, we can type in one or more values or use some more advanced options like parameter lists or macros.

In this example, we will keep it simple and just use a simple value of “Arizona”.

Press :”Save” and return to the burst details screen. Our value now appears.

Now we will create a distribution for this document.

Click on to go to the distribution screen:
Here we will store a PDF file on a network share and use a simple distribution macro to create a new folder using the name of the burst.

The Path is set to a fileshare and the [\$burst] macro means to replace this with the name of the burst, so in this example we will get a new folder under the “infoburst_reports” folder called “MyFirstBurst”.

As no specific filename is given, InfoBurst will keep the same name as the document (which in this case is “Sales per Region”).

If I wanted to store the file and call it “MyReport.pdf” I could specify the full-name like this

```
Path \wilcox\infoburst_reports\\$burst\MyReport.pdf
```

You will notice there are quite a few options for the network share, and there is a drop-down of other types such as Email, FTP etc but we will just keep it simple.

As for the format of the report, the default is PDF but if we wanted to choose another we could simply select the Format tab and choose another (like Excel or HTML).

Press “Save”, return to the details screen and press “Complete Burst” to see a summary of the burst we have created.
The burst exists and is ready to run. We can run it now by clicking on the “Execute Now” button. This is useful for testing as it does not require us to create a schedule.

When you Execute, you will be taken to the Today’s Schedule screen where you will see the burst start to run.

The icon will take you to the activity log, which is a step by step account of what the burst is doing (if it is running) or what it has done (if it’s finished).
Here you can see the steps, start a new session with BusinessObjects, logon, open, supply parameter, refresh etc.

An activity log is created for every burst that runs, and you can always review what happened using the **Logs->Activity Logs** menu.

**QuickBurst**

A facility exists to side-step some of the burst creation screens in order to quickly create a new burst that uses the most popular destinations (network share and email).

From the Catalog management screen, if you select one or more documents and press the button, you will see one single screen that allows you to enter any required parameters and choose from either Network or Email delivery (if you choose both, you will get Network delivery with Email notification).

Once the burst exists, you can edit using the standard screens (there is no way to edit a quick burst as it is now a regular burst).
Introduction to Report Bursting

In addition to supporting multiple documents within a burst, it is possible to use a single document with report bursting techniques in order to generate a set of outputs that are targeted by customer, region etc.

There are three methods that can be used to achieve report bursting:

1. Single-Pass
2. Multi-Pass
3. Combination of Multi-pass and Single-pass

Single-pass bursting is where there is just one data refresh and then a series of filters are applied to restrict what data is displayed. This option normally gives the best performance but does require some initial setup within the document to make this work.

Multi-pass bursting is where there are a number of data refreshes performed, typically using a dynamic database query to get the list of parameter values to use for each pass. This option works with any document type and does not require any special setup within the document itself.

For complex bursting it is quite typical to use both options together, for example to create a series of sales reports by state and city, a good solution might be to multi-pass refresh for each state and then single-pass burst on each city.

When a report is cataloged, InfoBurst discovers if there are any defined filters and therefore will use a single-pass.

Report Bursting is invoked when you supply more than one value for either a prompt or a filter. You can type in a hard coded list such as “Arizona;California” or more commonly you would use a parameter list that contains a list of values.

There are three types of parameter lists:

**IPL – Internal Parameter List**
This is suitable for small lists that do not change very often. The values are stored within the InfoBurst repository.

**EPL – External Parameter List**
This is a dynamic query or a file that will be resolved when a burst runs. It is suitable for larger value sets such as a list of product codes or customer accounts.

**RPL – Report Parameter List**
This is a list of values taken from the report itself.
As a simple example, we will now modify our “MyFirstBurst” to use a multi-pass to produce a report for Arizona and another for California.

Parameter *States*  
<table>
<thead>
<tr>
<th>Value or Value List (above)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arizona, California</td>
</tr>
</tbody>
</table>

Note we have checked the box to indicate we want a multi-pass. This checkbox appears if a prompt is detected that can take *multiple values*. If we wanted one report with both Arizona and California data, we would not check the box, but in this case we want a *separate* report for each value.

For the distribution, we now want to control the output so we will change the network path:

Path  
\`\\wilcoxit1\infoburst_reports\{$burst$}\=States\`

We will add another macro [=States] so that InfoBurst will now create a folder for each value of state, and again as no filename has been given we will get “Sales per Region.pdf” stored in each folder (one for “Arizona” and a second for “California”).

If we wanted, we could use a different path and create two files under the [$burst$] folder:

Path  
\`\\wilcoxit1\infoburst_reports\{$burst$}\Sales for [=States].pdf\`

This would create “Sales for Arizona.pdf” and “Sales for California.pdf” in the burst folder.

If we save, execute and view the activity log for the burst you will now see the two separate refreshes and the two files being published:
Burst Management in detail

The ‘burst’ is the fundamental unit within InfoBurst and there are many features and options that can be used.

The **Bursts->Burst Management** menu will display a list of existing bursts in each folder and you can create a new burst using either the **Bursts->Create New Burst** menu or from the catalog management screen.

If you wish to see a summary of a burst without having to edit the burst, click on the icon on the Burst Management screen to jump straight to the summary screen.

**Adding or removing documents**

On the Burst management screen, you can use the icon to edit the documents for a burst. Simply use the >> and << buttons to add or remove documents from the available documents.

It is possible to add the same document more than once (thus creating multiple instances). This may be useful if you wish to process the document more than once but with different parameters, however using report bursting techniques may offer a better solution.

**Changing the document name**

By default, the original cataloged document name will be used but you can change this name by clicking on the link (see below):

This will take you to the Document Options screen where you can change the name:
Setting document specific options

As above, if you click the document name you will see the Document Options screen where you can set options (if available):

**Refresh Report Data**

For BusinessObjects FullClient documents, you can select the data providers to refresh.

For all other documents you can choose if you wish to refresh report data.

**Ignore if Empty**

This option is supported for BusinessObjects FullClient, WebI and Crystal Reports.

If a report refresh returns no data, all distributions for the document will be ignored.

**Treat in-line a;b syntax as in-line IPL**

InfoBurst treats hard-coded parameter values such as “Arizona;California” as an internal parameter list. If you wish to disable this feature, un-check this option.

**Generate all combinations for multiple IPL/EPL**

If a document has multiple prompts/parameters and you use a list of values (IPL or EPL) for more than one prompt, InfoBurst will by default compute a matrix of all possible value combinations.

For example, if a document had a prompt for “Product” and also “Line” and product was set to use a list that generated “Product1” and “Product2” and Line was set to use a list that generated “Line1” and “Line2” we would get the following value pairs:

Product1 + Line1
Product1 + Line2
Product2 + Line1
Product2 + Line2

If you do not want this, and instead prefer to generate

Product1 + Line1
Product2 + Line2

.. then un-check this option and ensure that each list will generate the same exact number of values.
Deliver report only if SIBSDELIVER is true

This option will appear if delivery conditions have been created within the document.

See the next section (“Conditional Delivery using $SIBSDELIVER variable”) for details on how to set this up.

Enabled/Disabled/Disabled Next

Enabled means the document will be processed as normal.

Disabled means the document will not be processed until you re-enable.

Disabled Next means disable only for the next time it is used, and then re-set to enabled. This option can be useful if you are re-running a burst and wish to skip one or more documents temporarily.

Note: You can also set the enabled status for all documents in the burst by using the selector at the bottom of the burst details screen:

![All Enabled](image)

It is also possible to one-click toggle the enabled status by clicking on the status icon in the “Doc #” column.

Click on the green tick icon to toggle from Enabled to Disabled:

![Doc #](image)

Click on the red cross to toggle from Disabled to Disabled Next:
Clicking on the red warning will then toggle from Disabled Next to Enabled.
**Conditional Delivery using $IB$DELIVER variable**

It is possible to check a condition within a BusinessObjects FullClient document and based upon this condition; choose if you wish to distribute the report.

The condition needs to be created within the document as a variable that will evaluate to TRUE or FALSE.

Create a new variable with the special name $IB$DELIVER. It is important to name this variable correctly as InfoBurst will look for this exact name.

![Variable Editor](image)

Next, create a suitable formula that tests your condition and returns a 1 or 0. In the example below, we are using the **If** condition based around the **Sum** of a variable called `<Number of Orders>` and we are testing is this value is >1000. If it is, we return 1 else we return 0.
Save the document and re-catalog with InfoBurst so that the newly added condition will be detected. You can re-catalog by finding the document in the catalog management screen, click the edit icon to see the document details and then choose the Re-catalog action at the bottom of the screen.

When the document has been re-cataloged, you will see the following option on the document options screen:

- Deliver report only if $IB$DELIVER is TRUE?

Select this option to enable the delivery condition check.
The activity log will display a new `GET_VARIABLE` step for the `$IB$DELIVER` variable and if the option above has been enabled and the variable returns 0, the document will be ignored:

If the variable returns a 1 (or TRUE value), the distribution will continue as normal.
Using $IB$variables in document to allow conditions to vary

If the $IB$DELIVER condition may vary, for example in one burst you wish to test if the total number of orders is >1000 and in another burst the value might be 2000, you can create additional variables that can be set by InfoBurst. This also makes it easier to manage as you will not need to re-catalog or maintain multiple copies of the document.

To use a secondary variable, simply create the variable within BusinessObjects full-client with a prefix of $IBS$ and refer to this variable instead of a hard-coded value.
The formula should set an initial value:

![Variable Editor](image)

Edit the SIBSDELIVER formula and refer to the new variable:
The condition now tests for a value greater than \<$IB$VALUE> rather than a fixed value.

When setting the $IBSDELIVER condition inside the burst options, you will see also the variable(s) that can be set:

Here we are setting the value to 200, and also indicating that the variable is numeric. If you wish to assign a string, make sure to check the **String** type.
When running the burst, you will see a new step in the activity log that will `SET_VARIABLE` of any additional variables before the `GET_VALUE` of the `$IB$DELIVER` variable:

<table>
<thead>
<tr>
<th>Time</th>
<th>Step Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>10:42:57 AM</td>
<td>REFRESH</td>
<td>Sales per Region with Totals</td>
</tr>
<tr>
<td>10:42:59 AM</td>
<td>SET_VARIABLE</td>
<td>$IB$VALUE = 200</td>
</tr>
<tr>
<td>10:43:00 AM</td>
<td>GET_VARIABLE</td>
<td>$IB$DELIVER = 0</td>
</tr>
<tr>
<td>10:43:00 AM</td>
<td>IGNORE_DOCUMENT</td>
<td>Deliver Condition was not true</td>
</tr>
</tbody>
</table>

In the above example, a value of 200 caused the delivery condition to fail. Upon changing the `$IB$VALUE` to 100 and re-running the burst, we see:

<table>
<thead>
<tr>
<th>Time</th>
<th>Step Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>10:45:09 AM</td>
<td>REFRESH</td>
<td>Sales per Region with Totals</td>
</tr>
<tr>
<td>10:45:11 AM</td>
<td>SET_VARIABLE</td>
<td>$IB$VALUE = 100</td>
</tr>
<tr>
<td>10:45:11 AM</td>
<td>GET_VARIABLE</td>
<td>$IB$DELIVER = 1</td>
</tr>
<tr>
<td>10:45:11 AM</td>
<td>CREATE_PDF</td>
<td>Sales per Region with Totals.pdf</td>
</tr>
<tr>
<td>10:45:12 AM</td>
<td>STORE_NETWORK</td>
<td>/folder/InfoBurst_reports/Sales per Region with Totals.pdf</td>
</tr>
</tbody>
</table>
Supplying a document Parameter

If a document has parameters (or filters), you will see an icon in the Parameters column. Click on this icon to move to the document parameters screen.

A Parameter is a value used during the refresh of data. In BusinessObjects this is known as a prompt.

A Filter is a value used when report bursting using a single-pass approach (see section on Report Bursting).

A parameter or filter can be

1. A hard coded value such as “Arizona” or “1000”
2. A macro that is used to typically supply a date such as [=TODAY]
3. A list of values supplied by InfoBurst (IPL/EPL) to use with report bursting
4. A hard-coded “in-line” list such as “Arizona;California”
5. A fiscal calendar macro

In addition, a filter value can also be a list of data items that have been extracted from the report itself after a refresh (for example, refresh a report for a given date and then single-pass burst by each customer).

Multi-value parameters

In some documents (such as BusinessObjects) it is possible to create a parameter that can accept more than one value, meaning that you can create a report for both Arizona and California rather than a separate report for each state.

To create a multi-value prompt in BusinessObjects, you would use the operand “in list” rather than “equal to”.

If InfoBurst detects a multi-value parameter, you will see an extra option that allows you to treat a list of values as one value or to use a multi-pass approach and supply each value separately.

Value or Value List (ab:x)

Arizona;California

Refresh using each value (Multi-pass)
InfoBurst parameter Lists (IPL and EPL) to use report bursting

An IPL is a simple list of one or more values that you can create and maintain using the Admin->Report Bursting->Internal Parameter Lists menu.

An IPL is suitable for small lists that do not change often that do not warrant their own external data table in a database or file.

An EPL is a list that can be obtained from a file or more commonly a database query. Use the Admin->Report Bursting->External Parameter Lists menu to create and edit an EPL.

Click the icon to see the available lists that have been 'linked' to this parameter (see section on Report Bursting for more details).

An EPL is a good choice where the volume of data could be large and could change often. For example if you wish to report burst a document by each customer-id you could create an EPL using a database query to issue a query such as “select ID from customers where region=10” and InfoBurst will execute the query when the burst runs to build a list of customerID values, where it would use a multi-pass report burst approach to generate a report for each customer.

An EPL can also contain parameters, for example the query could be written like “select ID from customers where region=@region” and then when you select the EPL you will see something like “EPL:GetCustomers(@region=VALUE)” where you would replace VALUE with a suitable value.

Another more advanced example is where you wish to use an EPL to retrieve values and use a current parameter value as a parameter(!) – an example here would be a document with two prompts, State and ZipCode where you wish to retrieve a list of states using a regular IPL or EPL but for the ZipCode you want to use an EPL:GetZipCodes and pass the value of State.

If there were two states (such as Arizona and California), the EPL for GetZipCodes would be written like “EPL:GetZipCodes(@state=[=state])” where @state is the EPL parameter and [=state] is a macro that says “use the current value for state”, so when InfoBurst processes the document for State=Arizona, it will call the EPL:GetZipCodes(@state=Arizona) and then in the next pass for California it will call the EPL:GetZipCodes(@state=California).

Report Parameter Lists – RPL

An RPL is used to extract each data element from a report such as all customers, products or dealers. Typically you would assign an RPL to a filter when doing a single-pass burst.
**Creating a new distribution**

Select the ![icon](image) icon to create a new distribution for the document. You can create as many distributions per document as you wish.

Refer to the next chapter ‘Distribution options in detail’ for a complete explanation of the many options.

**Using a distribution template**

If you plan on using common formats, options or locations across multiple bursts or multiple documents you can save time and improve the manageability by creating and using ‘Distribution Templates’.

Use the **Admin->Distribution Management->Distribution Templates** menu to create a new template. You will first choose the document type/platform that the template will be available for and then you can create a distribution in the exact same way as a regular distribution.

To use/apply the template to a specific document in a burst, select the ![icon](image) icon to see a list of available templates. When you select a template from the list, it will appear under the distributions column.

**Applying a distribution template to multiple documents in the burst**

For a burst with more than one document, you will see an option at the bottom of the Burst Details screen that allows you to choose a template and apply it to all documents of the same type in the burst:

![icon](image)

For example if you have a template that stores in PDF format to a specific network share and you wish to do this with all Desktop Intelligence documents in the burst, you select the document type and press the ![icon](image) icon.
Global email options

Email distribution is handled at burst level rather than document level so that if during a burst execution a recipient is targeted to receive 3 files (say 2 PDF and 1 XLS) InfoBurst will attempt to combine these file attachments into a single email rather than send the files as separate attachments in 3 emails.

Note: The system configuration item EMAIL->MaxMessageSize is used to control the amount of data that can be attached to a single email. Check this value matches your email server or there may be errors if InfoBurst tries to send an email that is too large.

There are various options that you can set to control this ‘global’ email processing, to see these options check the “view global Email options” box at the bottom of the burst details screen:

Subject
This is the subject line for the generated email. If left blank, it will be set to ‘InfoBurst: xx” where xx is the name of the burst. Macros are supported so it is possible to set the subject to something like “Sales Reports for [=region]”.

From
This is the email address that the generated email will be sent from. Leave blank to use the default from the system configuration EMAIL->From.
Message

This is an optional additional message that will be combined with any document level email messages. It is typically used to provide general information such as how to contact the helpdesk or something similar. Macros are also supported.

You can choose if you wish the message to be added at the beginning of the email or at the end.

Allow recipients to un-subscribe to the reports in this burst

It selected, this will generate a message for the recipient that will allow them to elect not to receive these reports in the future. A hyperlink is added to the message that when clicked will log the user into InfoBurst as a guest account and ask them to confirm that they wish to un-subscribe.

After they confirm, they will receive an email that they can keep (like a receipt) and if they wish to re-subscribe in the future they can use the hyperlink to do this.

As an administrator, you can use the **System->Reinstate Email Subscriptions** menu to also ensure that the recipient will receive the reports for a specific burst.

Combine in-line and attachments where possible

If selected, InfoBurst will attempt to combine any in-line HTML reports with attachments. By default, in-line HTML email is delivered as a separate email due to the complexity of the report but in some cases (such as BusinessObjects FullClient) it is possible to do this.

Send ONE message with BCC list

This option is only available for bursts that contain a single document and is used when you wish to publish the same email to a large distribution list. Normally, InfoBurst will create and send a separate email to each recipient but this option allows one message to be sent to multiple recipients.
Email compression

The second tab on the global options is for setting up optional zip compression.

By default, attachments are not compressed but to save space and improve the time it can take to deliver large files you can choose to compress each file or compress all files into a single archive.

You can enter a password that must be used to open the archive and also elect to encrypt the data. If you encrypt data, the client will need WinZip v9 or greater to decrypt.
Distribution options in detail

Some destinations and formats will only be available if the document supports it, for example a BusinessObjects full client document will offer both plain text and CSV output but a Desktop Intelligence document on an XIR2 platform will only offer CSV.

The BusinessObjects Repository option will only be available to BusinessObjects 5.x/6.x and WebI 6.x documents.

The XI Platform option will be available to Desktop Intelligence, Web Intelligence and Crystal Reports that are stored on the platform and also for selected client/static documents if an EDP connection exists (this enables publishing to the XI Platform even for documents that are not originally stored there).

Each distribution is a combination of a destination and a format. The available destinations are:

- Network Share
- Email (or Email Notify + another destination)
- FTP
- InfoBurst Server
- WebDAV
- BusinessObjects Repository
- XI Platform
- Printer

The available formats are:

- Native
- PDF
- HTML
- Excel
- Text (Plain)
- Text (CSV)
- XML
Destinations in detail

Network Share

Previous Shares
This is a list of shares that have been entered before. Select one from the list to copy the share path into the Path where you can edit if required.

Path
This is the path and optional file-name where you wish to publish the document. You can use macros (see section on Macros) as placeholders for things like the name of the burst, the date in various formats, the values being used for report bursting etc.

Macros are enclosed in square brackets (e.g. [$burst] or [=State]) and allow you to fully control the path and/or file-name.

If a file-name is not given, InfoBurst will use the current Document Instance Name (the name of the document within the burst). This is typically the original name of the document when it was first cataloged but it can be changed.

If a macro is detected and you are report bursting, you must specify a file-name or you will risk over-writing output files.

InfoBurst will create any missing folders, this is typically done when report bursting whereby you could burst a report to a folder based on the current State/Customer etc.
Map to Share

The Path given must be available from the InfoBurst server, if it does not and you wish for InfoBurst to create the share dynamically, enter an NT user and password that will be used to map to the share.

Command Line

If you wish to invoke a user-controlled process such as a script or program, you can enter a command line here that will be called after the file(s) have been copied to the selected network path.

You can use the %B placeholder to pass the name of the burst and the %D placeholder to pass the name of the document.

If you wish for InfoBurst to wait for the process to complete before continuing, check the “Wait for complete” box.

Trigger File

A Trigger File is a simple text file that is created alongside the published document that can be used by external applications (typically in-house portal systems). The Trigger file name and content can be controlled using macros if required and default to the name of the document.

Create .zip

Use this option if you wish for InfoBurst to compress the document into a Zip. You can specify an optional password that must be used to open the Zip and also elect to encrypt the data. If you encrypt data, the client will need WinZip v9 or greater to decrypt.

Add Date/Time stamp

Use this option if you wish to add a simple date stamp to each published file. You can choose to place the stamp before or after the filename and also choose the format.
Email

You can select any combination of users/groups known to InfoBurst and also from any EDL (External Distribution Lists) or IPL (Internal Distribution Lists). If you wish to add email addresses manually, you can enter them into the input box and press “+” to add to the list.

An EDL is typically defined as a SQL query that can retrieve email addresses from a table (such as get all email for Sales Managers). When used in conjunction with report bursting InfoBurst will automatically pass along the current bursting parameters (such as State or Customer) which means each EDL query can use the parameter value in a WHERE clause to get email addresses for the current State/Customer.

To create an EDL, use the Admin->Distribution Management->External Distribution Lists menu.

Message

You can enter a message (that can include macros) that will appear in the email.

EDL Parameters

Some EDL queries might contain additional parameters, use this button to see the parameters and supply a value.

Attachment Name

By default an attached file will be named the same as the original document, however you can enter a new name if you wish. Macros can be used.
**FTP**

Type: **FTP**

<table>
<thead>
<tr>
<th>Previous Servers</th>
<th>[select a previously used server]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server</td>
<td></td>
</tr>
<tr>
<td>Remote Path</td>
<td></td>
</tr>
<tr>
<td>Logon</td>
<td>User: diuser</td>
</tr>
<tr>
<td></td>
<td>Password: ******</td>
</tr>
<tr>
<td>Trigger File</td>
<td>Name: [DOCNAME].GO</td>
</tr>
<tr>
<td></td>
<td>Content: [TODAY,MM/DD/YY]</td>
</tr>
<tr>
<td>Binary Mode ?</td>
<td>Yes</td>
</tr>
<tr>
<td>Connection Type</td>
<td>Normal ○ SSL ○ S/Shell</td>
</tr>
<tr>
<td>Port Numbers</td>
<td>Normal SSL 21 S/Shell 22</td>
</tr>
<tr>
<td>Create .zip ?</td>
<td>Yes [DOCNAME].zip</td>
</tr>
<tr>
<td></td>
<td>Password to open</td>
</tr>
<tr>
<td></td>
<td>Encrypt data (128-bit AES) ? Yes [requires password]</td>
</tr>
<tr>
<td>Add DateTime stamp ?</td>
<td>No ○ Suffix ○ Prefix Format [MM_YY]</td>
</tr>
</tbody>
</table>

**Previous Servers**

This is a list of servers that have been entered before. Select one from the list to copy the server name into the **Server** where you can edit if required.

**Server**

This is the name or address of the FTP server.

**Remote Path**

This is the path and optional file-name where you wish to publish the document. You can use macros (see section on Macros) as placeholders for things like the name of the burst, the date in various formats, the values being used for report bursting etc.

Macros are enclosed in square brackets (e.g. \[$burst\] or \[=State\]) and allow you to fully control the path and/or file-name.

If a file-name is not given, InfoBurst will use the current Document Instance Name (the name of the document within the burst). This is typically the original name of the document when it was first cataloged but it can be changed.
If a macro is detected and you are report bursting, you must specify a file-name or you will risk over-writing output files.

InfoBurst will create any missing folders, this is typically done when report bursting whereby you could burst a report to a folder based on the current State/Customer etc.

**Logon**

Enter the user name and password to establish the connection with the FTP server.

**Trigger File**

A Trigger File is a simple text file that is created alongside the published document that can be used by external applications (typically in-house portal systems). The Trigger file name and content can be controlled using macros if required and default to the name of the document with today’s date inside.

**Binary Mode**

Check this option if you are transferring binary data such as PDF.

**Connection Type**

Normal mode is standard FTP.

SSL is secure FTP using the FTPS protocol over a secure socket connection.

S/Shell is secure FTP using the SFTP protocol.

**Port Numbers**

For Normal or SSL mode, enter the port number (default is 21).

For S/Shell mode, enter the port number (default is 22).

**Create .zip**

Use this option if you wish for InfoBurst to compress the document into a Zip. You can specify an optional password that must be used to open the Zip and also elect to encrypt the data. If you encrypt data, the client will need WinZip v9 or greater to decrypt.

**Add Date/Time stamp**

Use this option if you wish to add a simple date stamp to each published file. You can choose to place the stamp before or after the filename and also choose the format.
InfoBurst Server

Type : InfoBurst Server

# Versions

The InfoBurst server destination is where the document is published centrally to the InfoBurst website and can be viewed using Documents->My Server Documents.

# Versions

If left blank, older versions will not be removed and can be purged using Housekeeping functions.

If a value is given, InfoBurst will automatically remove old versions of documents.

Note: InfoBurst will create a version for every combination of a document within a burst.

For example if a prompted documented called “Sales Summary” was run once a week from a burst called “Weekly Sales” with #Versions=4 and the same document was run once a month from a burst called “Monthly Sales” with #Versions=12, then InfoBurst would ensure that the two “sets” of documents were treated separately.
WebDAV

WebDAV allows you to publish to web folders and Sharepoint sites.

**Previous Shares**

This is a list of shares that have been entered before. Select one from the list to copy the share path into the **Path** where you can edit if required.

**Path**

This is the http:// path and optional file-name where you wish to publish the document. You can use macros (see section on Macros) as placeholders for things like the name of the burst, the date in various formats, the values being used for report bursting etc.

Macros are enclosed in square brackets (e.g. [$burst] or [=State]) and allow you to fully control the path and/or file-name.

If a file-name is not given, InfoBurst will use the current Document Instance Name (the name of the document within the burst). This is typically the original name of the document when it was first cataloged but it can be changed.

If a macro is detected and you are report bursting, **you must specify a file-name** or you will risk over-writing output files.

InfoBurst will create any missing folders, this is typically done when report bursting whereby you could burst a report to a folder based on the current State/Customer etc.
**Logon**

Enter the user name and password to establish the connection with the shared folder.

The user value can contain “DOMAIN\User” if required.

**Create .zip**

Use this option if you wish for InfoBurst to compress the document into a Zip. You can specify an optional password that must be used to open the Zip and also elect to encrypt the data. If you encrypt data, the client will need WinZip v9 or greater to decrypt.

**Add Date/Time stamp**

Use this option if you wish to add a simple date stamp to each published file. You can choose to place the stamp before or after the filename and also choose the format.
**BusinessObjects Repository**

This option is available for BusinessObjects full-client documents.

If a WebI 6.x connection is available (created with the **System->Enterprise Document Providers** menu) there are some additional browse options to assist when selecting users, groups and category names and it is also possible to publish PDF/Excel output from a full-client document (and static documents too).

### Platform

For a full-client document in Native format, this does not apply.

If one or more WebI 6.x connections exist you will be able to specify PDF or Excel format for the full-client document, and you will see a platform selector:

![Platform Selector](image)

### Document Name

This is the unique name of the document to be stored in the repository.

You can use macros (see section on Macros) as placeholders and would typically only do so when using report bursting.
Group(s) or Users(s)

To send to one or more groups (defined in Supervisor), enter the Group names separated by “;” for example “Sales;Marketing”.

To send to one or more users (defined in Supervisor), enter the User names separated by “;” for example “John;Steve” and check the option “Send to user”.

If a WebI 6.x connection exists, you will be able to browse for users or groups by clicking on these icons:

- Browse for Groups
- Browse for Users

When sending to a user, it is also possible to use an EDL or IPL to generate a list of users (this is the same technique as sending Email using a dynamic list). To specify a list, click on the icon and select a list.

Category

Enter one or more category names separated by “;”.

If a WebI 6.x connection exists, you will be able to browse for categories by clicking on the icon.

Document Domain

Select from the drop-down of available domain names. This list is built using the BOInit utility that normally runs when InfoBurst is first installed. If a document domain is not listed, ask your administrator to add the domain by re-running BOInit.

Overwrite if exists

Check this option to allow InfoBurst to over-write a document in the repository if it already exists. If this option is not used and the document already exists, this step of the burst will abort.

Add Date/Time stamp

Use this option if you wish to add a simple date stamp to each published file. You can choose to place the stamp before or after the filename and also choose the format.
**XI Platform**

**Folder**

Enter a folder name such as “Sales” or “Sales\Weekly Reports” or use the icon to browse the available folders.

You can use macros (see section on Macros) as placeholders and would typically only do so when using report bursting.

InfoBurst will create any missing folders, this is typically done when report bursting whereby you could burst a report to a folder based on the current State/Customer etc.

**Document Name**

This is the name of the document to be stored in the selected folder.

You can use macros (see section on Macros) as placeholders and would typically only do so when using report bursting.

**Overwrite if exists**

Check this option to allow InfoBurst to over-write a document on the XI platform if it already exists. If this option is not used and the document already exists, this step of the burst will abort.

**Add Date/Time stamp**

Use this option if you wish to add a simple date stamp to each published file. You can choose to place the stamp before or after the filename and also choose the format
**Printer**

This option is available for BusinessObjects full-client documents and Crystal Reports full-client documents.

The list of available printers is managed by the IBRegister utility. If the printer name you need is not in the list, ask your administrator to configure the printer for use on the InfoBurst server and re-run the IBRegister utility.


**Formats in detail**

You can choose from a variety of different formats and depending on the document type, certain more specific options may be available.

The following icons will be used to denote that an option is available only for a specific document type:

- BusinessObjects FullClient
- BusinessObjects Web Intelligence 6.x
- Crystal Reports 9.x, 10.x or 11.x
- BusinessObjects Desktop Intelligence on XIR2
- BusinessObjects Web Intelligence on XI platform
- Crystal Reports on XI platform

**Native**

This format keeps the report in the original format, so a BusinessObjects FullClient document will be stored as a .rep file and a Crystal Reports document will be stored as a .rpt file.

**Save for ALL users**

Check this option to save the document so that it can be opened by any user.

**PDF**

This format will create an adobe PDF output that is suitable for most destinations (network, email etc) and is a good format for general viewing and printing.

**Split by Report-Tab**

Check this option if you wish to generate a separate PDF for one or more report-tabs in the document rather than one PDF for the whole document.

Use the **Reports** tab to select the report-tabs.
**HTML**

This format will create an HTML output that is suitable for a network share (e.g. a website), InfoBurst server or email.

For email you can choose if you wish to “in-line” the content whereby InfoBurst will embed the report and any images directly into the email rather than attach any files.

For BusinessObjects/WebI documents, you can use the **Reports** tab to choose the particular report-tabs to include.

**Split by Section**

Creates a separate output for each section in the report.

**If Email, deliver as separate reports**

For Email distribution, will deliver each report-tab as a separate email.

*Note*: For WebI documents, this option is always used.

**If Email, deliver as in-line**

For Email distribution, will embed the content and images into the body of the email message.

**Auto Refresh page every [n] seconds**

This option adds a refresh interval to the resulting HTML output so that the page will re-load every [n] seconds. This option is useful if the HTML is being displayed all the time and the data is being refreshed every [n] minutes.

**If not Email, create files in a root folder**

For Crystal Reports and non-Email distribution, this option determines if files are grouped together in a root folder of the same name as the document.
**If Network, generate one file**

For Network distribution, will create a single .htm file rather than the default folder structure. This option will only be available for FullClient documents that have a single report-tab and any graphs or images will not be available.

There are two additional options that can be used to remove selected HTML markup if you are planning to integrate the content into your own portal system:

**Remove HEAD section**

This will remove all content in the page header section.

**Remove outer BODY tags**

This will remove the <BODY> tags around the main page content.
**Excel**

This format will create an Excel workbook that is suitable for most destinations (network, email etc) and is a good format if you wish to manipulate the data.

For BusinessObjects 5.x FullClient documents, InfoBurst can convert the document via HTML into Excel and offer the ability to further customize the workbook by using a ‘base workbook’ and a custom macro.

For BusinessObjects 6.x and Desktop Intelligence, InfoBurst can also convert via HTML but also offers the native export provided by BusinessObjects.

If you choose InfoBurst Conversion, you can use the **Reports** tab to choose the particular report-tabs to include. For Desktop Intelligence and Web Intelligence documents on the XIR2 platform you will also have an option to split by report-tab.

A ‘base workbook’ is a standard Excel workbook that has been cataloged as a static document within InfoBurst. This workbook can then be used as a starting point whereby InfoBurst can add the tabs from the source document and optionally run a macro that could perform specific tasks before the workbook is distributed.

For most other document types, InfoBurst offers the ability to create the Excel output using the standard export option of the document provider and if you have cataloged one or more Excel workbooks, you will have the option to perform ‘post processing’ which allows you to further customize the workbook.

**InfoBurst Conversion / Provider Conversion with Post Processing**

**Base Workbook**

Choose from the list of cataloged Excel workbooks. InfoBurst will use this workbook as the base instead of an empty workbook.

**Macro to run (in base)**

Enter the name of an existing macro that you want InfoBurst to execute. **Note:** It is important that you take extreme care when running a macro as an error could result in Excel stopping and causing the burst to hang.

Refer to the section in this guide about using Excel macros for more details.
Remove Base Tabs

Check this option to ensure that any existing tabs in the base workbook are removed. If you wish to keep selected tabs, enter their names (separated with “;”;).

Autofit cells

Check this option to apply an auto-fit to all cells in the workbook.
Text (Plain)

This format will create a text representation of the document (not the raw data, see CSV).

For BusinessObjects documents, you can use the Reports tab to choose the particular report-tabs to include.

File Extension

Enter the name of the extension to use. The default is “.txt”.

Replace TAB

If you wish to replace any TABS (CHR(9)), enter a character to use instead.

Generate Doc.txt rather than Doc_Report.txt

For BusinessObjects reports with a single report-tab or a single selected tab, check this option to generate a file based only on the document name (will not append the tab-name).

If Email, deliver as in-line

Check this option to embed the text of the report into the body of the email.
**Text (CSV)**

This format will generate the raw data behind the report in comma separated value format.

For BusinessObjects/WebI documents, one file will be generated per data provider.

**File Extension**

Enter the name of the extension to use. The default is “.csv”.

**Delimiter**

Enter the character to use for a data delimiter. The default is “,”.

To use tabs as a delimiter, press the “Tab Delimited” button.

**Header Line**

Check this option if you wish to remove the first line of the generated file which is typically a header line.
XML

This format will generate an XML representation of the document.

For BusinessObjects/WebI documents, you can use the **Reports** tab to choose the particular report-tabs to include.

For BusinessObjects 6.x, Desktop Intelligence and Web Intelligence XIR2, it is also possible to create XML based on the data in each Data Provider.

If the document has more than one Data Provider, you will be able to choose which ones you wish to export.

If a single Data Provider is selected, the output file will be name Document.Xml where Document is the name of the document instance within the burst.

If more than one Data Provider is selected, a separate .xml output file will be generated for each provider and the name of the Data Provider will be appended to each file.
**Email Notification**

For all destinations except for Email itself, you can choose to send an email to notify recipients that reports have been published and provide a link to the report.

You can choose the recipients in the same way as regular Email (users, groups, dynamic lists, ad-hoc addresses).

**Message**

You can enter a message (that can include macros) that will appear in the email.

**EDL Parameters**

Some EDL queries might contain additional parameters, use this button to see the parameters and supply a value.

**Include Link(s)**

Check this option to add a link to the report so the user can click a link and go straight to the report. The default is to include the link.
Scheduling in detail

This chapter explains how to create schedules that fit into your production workflow and how to monitor and change burst activity.

What is an InfoBurst Schedule?

An InfoBurst Schedule consists of one or more Bursts. A Burst is the unit of execution NOT the Schedule. Bursts run concurrently on the InfoBurst Server up to the limit configured within InfoBurst (MaxNumber). InfoBurst Schedules are managed by the InfoBurst Scheduler, a fully featured scheduler that is built-in to the product.

24-hour Daily Schedule

This refers to the list of bursts that are scheduled for the current 24-hour period. Any changes to a schedule will automatically be reflected and corresponding bursts added to, moved or removed from Today’s schedule accordingly.

How does the InfoBurst scheduler work?

The Scheduler runs as a System Service under Windows, and is automatically restarted if the Server is re-booted. It is known as IBSERVER – the “InfoBurst Cataloging and Scheduling Service”. Under normal circumstances, it will be running at all times.

It performs THREE vital jobs within InfoBurst: -

1) At midnight, the start of a new day, it schedules ‘bursts’ for execution within the next 24-hour period, and these are viewable under the “See Today’s Schedule” menu option in InfoBurst.

2) When a Burst is ‘Ready’ to execute (it has satisfied its Date & Time and/or Event constraints), it dispatches control to an instance of IBRUN, a separate process, to perform the execution of the burst. IBRUN communicates with IBSERVER in order to inform the scheduler when the burst has completed.

3) It performs Document Cataloguing and interrogates Provider Instances such as Business Objects to retrieve document / parameter information.

All work carried out by the scheduler is logged in the ‘Server Log’. Your Administrator can interrogate the server log from the Web Application or by going directly to the log file itself.
**What can be scheduled?**

There are three types of process that can be scheduled: -

(A) Standard InfoBurst ‘Bursts’ – logical groups of documents for refresh & distribution.

(B) External Data Source (XDS) refreshes – automatic refreshing of data that is used to populate a Crystal Xcelsius dashboard.

(C) System Housekeeping tasks – created as a ‘System Schedule’ it is possible to define tasks that are executed at regular intervals to perform deletion of old system log files.

To create a new schedule of Standard InfoBurst ‘Bursts’ (as in A above), select ‘Schedules’ from the menu and ‘Create New Schedule’

![Create New Schedule](image)

**Schedule options**

You are prompted to enter the details for the schedule. Use the tabs provided to switch between screens.

<table>
<thead>
<tr>
<th>Description</th>
<th>New Schedule</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enabled</td>
<td>Yes</td>
</tr>
<tr>
<td>Priority</td>
<td>10</td>
</tr>
<tr>
<td>Always Valid ?</td>
<td>To run this schedule between certain dates.</td>
</tr>
</tbody>
</table>
Enabled
This option means that you can temporarily disable a schedule. The schedule will not be a candidate for processing until it is re-enabled.

Priority
Use this if you wish to assign a priority to a schedule other than the default of 10 (the lowest). When several schedules hit the system at the same time, a schedule with a higher priority (towards priority 1 the highest) will queue its bursts before those bursts in a lower priority schedule.

Validity
The default value for this is ‘always’ – the schedule will always be a candidate for processing. Set a validity date range if you wish to run this schedule for a limited period only. E.g. Just for the month of December.

Folder
This is where the schedule will be stored in your folder structure. The default is ‘My Schedules’ your private folder. You can store your schedule in a Public or a Shared folder as an alternative, if you need it to be accessible by others.
Double-click on a burst or use the >> chevrons to select a burst for this schedule

**Frequency**
There are a wide range of frequency options:

- Daily
- Once
- Hourly
- Every
- Monthly
- On certain dates
- When events are set

Once/Hourly/Every/Daily/Monthly
These cater for the majority of scheduling requirements

On certain dates/Using a calendar
These allow you to choose one or more specific dates, or to define a custom calendar containing rules that allow you to define a very specific schedule calendar for your organization. Using the Custom Calendar, you can view the calendar into the future to easily see the impact of your scheduling rules in subsequent fiscal or calendar years.
Using a Calendar

Using InfoBurst you are able to create one or more custom calendars that reflect your business schedule dates. Use calendar ‘rules’ and define fixed dates to comprise your calendar.

When events are set
You can define ‘events’ that must be met before a schedule will run. This means that you can define schedules that are conditional. See ‘Events’ below for an explanation of the event types supported in InfoBurst.
Events
Events are defined from the ‘Manage Events’ menu item. An event is normally linked to just one schedule; however, a schedule may be linked to several events. All events linked to a schedule must be satisfied before the schedule will run. There are three event types:

- **File Exists**
  - A file must exist on the network before the event is satisfied.

- **Boolean Trigger**
  - The Boolean value for the trigger must be ‘true’ before the event is satisfied.

- **DB Query**
  - A value returned by a SQL query must match the expected value.
  
  To define a database query, first create a DB Connection entry and test it.

![Database Connection Example](image)
InfoBurst allows you to reset events automatically or manually. If Automatic is specified events are reset using the following methods:

- **File Exists**: the file is deleted from the network
- **Boolean Trigger**: the trigger is set to false
- **DB Query**: you supply an ‘Update’ SQL statement that will execute

### Example Configuration

<table>
<thead>
<tr>
<th>Event Name</th>
<th>Type of Event</th>
<th>Reset</th>
</tr>
</thead>
<tbody>
<tr>
<td>Datamart_Sql</td>
<td>DB Connection</td>
<td>Automatically after use</td>
</tr>
<tr>
<td></td>
<td>DB_MyDataMart</td>
<td>Manually</td>
</tr>
<tr>
<td></td>
<td>SQL Query</td>
<td></td>
</tr>
<tr>
<td></td>
<td>select [status] from [Statuslookup]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SQL for AutoReset</td>
<td>update [Statuslookup] insert 'PENDING' into [status]</td>
</tr>
<tr>
<td></td>
<td>Event is TRUE when this value</td>
<td>READY</td>
</tr>
<tr>
<td></td>
<td>Test</td>
<td></td>
</tr>
</tbody>
</table>

*Note: For FileExist events, AutoReset means “delete file”*
At End actions
There are two types of at end action:

At Normal Completion
You can send an email and/or set a Boolean trigger.

Upon 1 or more burst aborts
You can send an email and/or set a Boolean trigger.

In both cases, the sending of an email requires you to create and associate an email template with this request.
**Monitoring the Status of a Burst Execution**

Each Burst invoked from a Schedule has a Status attaching to it. The possible values are listed below in the sequence in which they normally occur:

<table>
<thead>
<tr>
<th>Status</th>
<th>Description</th>
<th>Valid Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>PENDING</td>
<td>Due to be executed later today</td>
<td>HOLD, FORCE, DELETE</td>
</tr>
<tr>
<td>MISSED</td>
<td>Not executed because of a Service Interruption</td>
<td>FORCE, DELETE</td>
</tr>
<tr>
<td>READY</td>
<td>Ready to execute (Scheduling constraints satisfied - Date &amp; Time and/or Events) BUT waiting for an available instance of IBRUN to execute the burst</td>
<td>HOLD, DELETE</td>
</tr>
<tr>
<td>WAIT EVENT</td>
<td>Waiting for an Event – either a TRIGGER and/or a FILE_EXISTS condition to be satisfied – in order to begin execution</td>
<td>DELETE</td>
</tr>
<tr>
<td>HELD</td>
<td>Burst Held due to the Administrator issuing the HOLD action from the Today’s Schedule screen</td>
<td>RELEASE, DELETE</td>
</tr>
<tr>
<td>EXECUTING</td>
<td>Burst is executing under the control of an instance of IBRUN</td>
<td>STOP, KILL</td>
</tr>
<tr>
<td>STOPPING</td>
<td>InfoBurst has detected an Operator STOP request, but has so far not managed to communicate with the instance of IBRUN to carry out the request</td>
<td>KILL</td>
</tr>
<tr>
<td>STOPPED</td>
<td>Burst Stopped by the Administrator issuing the STOP action from the Today’s Schedule screen</td>
<td></td>
</tr>
<tr>
<td>COMPLETED</td>
<td>Burst completed OK</td>
<td></td>
</tr>
<tr>
<td>ABORTED</td>
<td>Burst aborted</td>
<td></td>
</tr>
<tr>
<td>WAIT KILL</td>
<td>Kill request has been submitted</td>
<td></td>
</tr>
</tbody>
</table>

**Calendars**

It is now possible to build a custom calendar that reflects the special scheduling requirements for your organization. Once built, it is possible to copy and re-use the calendar and modify it for use in subsequent years. You can build a Custom Calendar based upon a regular Calendar Year or your Company Fiscal Year.
**Viewing a future schedule**

This option enables you to view bursts that are ‘scheduled’ to run on a future date. E.g. If you build a weekly schedule to execute on a Monday, you will see that the bursts show up every Monday into the future.

**Using an external scheduler**

If you already have or prefer to use a scheduler of your own, then it is possible to interface to the ‘Bursts’ defined in InfoBurst and execute and monitor them. There are two commands that are available to do this:

IBEXEC - execute a burst & conform to scheduling constraints of InfoBurst
You would run this command from the InfoBurst server.

IBRMT - execute a burst & conform to scheduling constraints of InfoBurst
You would use this command from a server remote from InfoBurst.

These commands are explained in detail in the InfoBurst Installation & Configuration Guide.
Report Bursting in detail

Report Bursting is a very powerful feature and can be used to deliver specific content in a variety of different ways.

Some examples of report bursting have already been shown (see Introduction and Getting Started) but this chapter fills in the gaps and explains how to setup documents for single-pass bursting and how to establish links between IPL/EPL and report parameters.

Key Concepts

As described in the Getting Started chapter, there are two approaches when it comes to bursting a report.

Single-pass bursting is where there is just one data refresh and then a series of filters are applied to restrict what data is displayed. This option normally gives the best performance but does require some initial setup within the document to make this work.

Multi-pass bursting is where there are a number of data refreshes performed, typically using a dynamic database query to get the list of parameter values to use for each pass. This option works with any document type and does not require any special setup within the document itself.

In short, report bursting means ‘driving’ a report with a list of values such as a list of customers or a list of products rather than a single value.

When you are bursting, you are iterating though the list of parameter values and for each one (or each set of values) InfoBurst will generate the output as per the distribution options and deliver the report.

For example, if we burst a report by State, we are going to create a separate report for each state and we need to take care that we create unique output files so that we don’t overwrite anything. This is where InfoBurst macros like [=State] can be used to control the file-name or folder.

By default, if you do not use ANY macros in a distribution, InfoBurst will generate a simple report name and concatenate each report parameter to ensure a unique name is used.
For example, if we burst a document called “SalesByState” by State in PDF format using an IPL that contains “Arizona” and “California” with a network delivery to a folder such as “\reports\weekly\SalesReports” InfoBurst will generate the following files in the given folder:

- SalesByState_Arizona.pdf
- SalesByState_California.pdf

If you use ANY macro, InfoBurst will assume that you have taken control over the path and will NOT generate a default name.

For example, if you wish to store each report in a folder of each state name, you could use the path

\reports\weekly\SalesReports\[=State]

This will create the following:

- SalesReports\Arizona\SalesByState.pdf
- SalesReports\California\SalesByState.pdf

You can use the macro [=State] as part of the folder-name, file-name or both. You can use any number of macros in any you wish. For example

\reports\weekly\SalesReports\[MM]_\[DD]\Report for [=State].pdf

This uses the simple [MM] macro to generate the month number and [DD] to generate the day number and will create the following (assuming today is 11th October):

- SalesReports\10_11\Report for Arizona.pdf
- SalesReports\10_11\Report for California.pdf
**Creating and Linking parameter lists**

**IPL – Internal Parameter List**

An IPL contains a simple list of values that you can manage through the web interface.

Use the **Admin->Report Bursting->Internal Parameter Lists** menu to manage IPL’s.

Simply add one or more values using the input area. To quickly add more than one value, separate each value with the delimiter shown at the left-side of the input area. You can change this delimiter if you wish.

![Input area with values: One, Two, Three]

Use the “+” button to add the values and the “-“ button to remove a selected value.

By default, the list will be sorted but you can un-check “Sort List” if you wish.

Normally an IPL would be used for report bursting to supply a list of values to a parameter (such as a list of regions) but it is also possible to create a list of values that will be used as part of a **conditional distribution**.

An example of a conditional distribution could be if you were bursting by state and were creating a PDF report for every state but you also wish to publish an Excel version for selected states, you could add the states to an IPL list and check the option “**Use only for distribution condition**” and this IPL will not appear as an option for bursting but will appear as an option for a condition linked to the parameter.

Each IPL that you create needs to be “linked” to one or more “Master Parameters” so that it is available for selection on the burst parameters screen. See the section “Linking an IPL/EPL to a Master Parameter”.

---

**InfoSol™**

75  
v2.5 - October 2006
EPL – External Parameter List

An EPL can dynamically read values from an external file or more commonly execute a SQL query against a database table.

Use the Admin->Report Bursting->External Parameter Lists menu to manage EPL’s.

From File

Enter the path to the input file and if this is a UNC path that does not already exist on the InfoBurst server you can enter an NT user/password that will be used to map to the share.

From Database Query

Before you can use this option, you will need to configure at least one connection to an external database using the Admin->DB Connections menu.

Choose the connection and enter a SQL query that you wish to execute. If you wish to use parameters (such as a WHERE condition that you want to vary based on a value) you can use the @ to name a parameter.

Ensure that the Result Column Name is set correctly as this is the column that will be used to get the value.

For a string value, we recommend that you enclose the value in single quotes like this:

Select ProductName from MasterProducts where ProdCategory='@category'
Normally an IPL would be used for report bursting to supply a list of values to a parameter (such as a list of regions) but it is also possible to create a list of values that will be used as part of a **conditional distribution**.

An example of a conditional distribution could be if you were bursting by state and were creating a PDF report for every state but you also wish to publish an Excel version for selected states, you could add the states to an IPL list and check the option **“Use only for distribution condition”** and this IPL will not appear as an option for bursting but will appear as an option for a condition linked to the parameter.

**Linking an IPL/EPL to a Master Parameter**

The idea behind “Master Parameters” is that a single list (such as a list of states) can be used with any parameter/prompt that is related to the list, and quite often report authors will use a different prompt string!

For example, all of these prompts are asking for a state:

- State
- States
- Enter a State
- Enter one or more States

When a document is cataloged with InfoBurst, the discovery phase will identify any prompts and InfoBurst keeps track of the unique names across all documents in the system.

The advantage to this approach is that you can create one list of states and “link” it to all prompts that relate to “State”.

You can of course create as many lists as you wish and link them as you need, for example you might in the case of states create a list of “Western States” and “Eastern States” if you are report bursting by region.

To link the list, simply check one or more of the Master Parameters in the list

```plaintext
  STATE
  STATES
  USER
```
**Multi-pass Bursting**

There is no special setup required for any document in order to use multi-pass bursting. Simply supply a list of values using either a hard-coded string such as “Arizona;California” to a prompt or more typically create and use either an IPL or EPL to generate a list of values.

**Single-pass Bursting using a Filter**

To use single-pass bursting requires that a Filter be available that InfoBurst can use to be able to split the report into pieces.

InfoBurst uses a common approach to identify a filter in a report, the filter variable must begin with the prefix #IB# for example #IB#Customer.

**Setting up a filter for BusinessObjects FullClient**

Open the document and create a new variable with a name prefixed #IB#: 

![Variable Editor](image-url)
Set the formula to something simple such as ="" or a valid value like "=85027".

Next, set up a global filter for each report-tab where you will specify the data element in the report to filter and set the formula to include the newly created variable:

Click Add to create a new filter and select the data item to filter:

Next, click on Define to set up the formula:
Click first in the variables list and identify the data element (in this case Customer Zipcode) and then type = and select the #IB# variable (in this case #IB#ZipCode).

The formula should be

\[ \text{Data Element} = \#IB\#Variable \]

It is very important that you get this formula the right way round!

**Save the document and re-catalog with InfoBurst** so that the newly added filter will be detected. You can re-catalog by finding the document in the catalog management screen, click the edit icon to see the document details and then choose the **Re-catalog** action at the bottom of the screen.

Once the document has been re-cataloged, you can check the document details again and you should see on the Parameters tab there is now a new Filter that matches the name of your #IB#variable.
Setting up a filter for BusinessObjects WebI

You do not need to change the document in any way, InfoBurst can create a dynamic filter.

First, find the document using catalog management and click to edit. Go to the Variables tab and you will see a list of data elements and a checkbox for “Use as Filter”.

Select the data element and press Save. You will now see the filter has been added to the Parameters tab.

Note: If you add a filter to a WebI document that has already been used in a burst, you will see a message:

This message informs you that the filter has also been added to any existing bursts with a value of “*” so that there is no side-effect from adding the filter (“*” means all values).
Setting up a filter for Crystal Reports

Open the document and create a new report parameter the prefix #IB# (for example #IB#Region) and ensure this parameter can accept multiple values.

Use the Select Expert as per any parameter and connect the parameter with the “is equal to” function.

Save the document and re-catalog with InfoBurst so that the newly added filter will be detected.
Using an RPL (Report Parameter List) with a Filter

An RPL is a list of values extracted from the report itself, for example **RPL:Customer City** would extract each unique value for the data element “Customer City”.

When supplying the value for a **Filter** (on the Document Parameters screen) you will see all available Report Parameter Lists for the document when you click on the \( \equiv \) icon:

Combining both Multi-pass and Single-pass bursting

Often you will need to find balance between performance and flexibility and therefore it is quite common to use both multi-pass and single-pass together.

For example, you could add a filter \#IB#City to a report that also has a parameter for State and then multi-pass the State using an IPL or EPL (so we refresh the report per state) and then for the filter City you could use an RPL to split the report for each City.
Looking up a report element when single-pass bursting

If you are bursting a document on a value such as Customer, it is possible to retrieve an associated value such as ZipCode even though ZipCode is not a parameter or filter.

This feature is available only for BusinessObjects FullClient documents and will only work if the filter variable (#IB#Customer) matches the underlying data element in the document.

For example, if you create a filter called #IB#Customer and the underlying global filter is set to operate on a data element called <Customer> then InfoBurst will be able to retrieve associated values for each Customer.

If however you created the #IB#Customer filter using an underlying data element called <Customer Name> there is a possibility that InfoBurst will not be able to find associated items reliably.

An example of when you might use this feature would be if you wish to burst a document by customer and store a PDF on the network in a folder based on the ZipCode for the customer.

The macro [=ZipCode] would then be used to retrieve the current value of ZipCode for the current value of the filter.
Conditional Distribution

Whenever you have a list of values being supplied (e.g. a list of States) you can use a powerful feature called conditional distribution to only deliver in certain locations/formats IF a condition is true (e.g. State=Arizona).

On the Burst Details screen, when you create a distribution that uses multiple input values you will see an icon next to the distribution:

![Distribution Icon](image1)

If you click on this icon, you will see a panel appear that allows you to choose if there any conditions that must be met for this distribution to be valid:

![Conditions for Distribution Panel](image2)

For each parameter or filter that has a value list, you can choose either that the current value must be in another list (IPL or EPL) or must be a specific value.

It is possible to create a list (IPL or EPL) specifically for use as a distribution condition (see earlier section on Creating and Linking Parameter Lists).

Once a condition has been set, the icon will change color to indicate a condition exists:
Email in detail

This chapter explains the various methods that can be used to send reports via email.

Selecting Email Recipients
Currently there are 10 ways to target a report to potential email recipients:

- **InfoBurst User**
  Target a User who is registered within InfoBurst

- **InfoBurst Group**
  Target a group of Users registered within InfoBurst
  E.g. the InfoBurst Support Group

- **Internal Distribution List**
  Target a list of recipients that you have built within InfoBurst (IDL)

- **External Distribution List (File list)**
  Target a list of recipients that you have created on an external file (EDL)

- **External Distribution List (LDAP connection to email server)**
  Target a list of recipients by querying your email server with search criteria

- **External Distribution List (DB Query)**
  Target a list of recipients by running a SQL query against a Database (EDL)

- **External Distribution List (DB Query with filtering)**
  Target a list of recipients by running a SQL query against a Database (EDL) and including where conditions to filter out a specific set of people

- **Ad-hoc email address**
  Enter one or more ad-hoc email addresses for this distribution

- **Group email address (resolved on email server)**
  Enter a group email address that will be resolved by your email server and result in a logical set of recipients

- **Email address contained in BO report (implicit distribution list)**
  Retrieve the email addresses stored in a BO report for use with distribution

- **Email address contained in BO report (from Excel data provider)**
  Retrieve the email addresses resulting from a BO query against a Microsoft Excel data source defined for the report

Note: Some of these are described in more detail later in this section

It is possible to combine the following option:

- **Email Un-subscribe** (to allow email recipients to un-subscribe from a report)

In addition – for a distribution that is not via email:

- **Email notification** (all distribution types - include or exclude report links)
Email Recipients from **Internal Distribution List**

Select Admin / Distribution Management / Internal Distribution Lists

Create / Edit a list

<table>
<thead>
<tr>
<th>Select</th>
<th>List Name</th>
<th># Values</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>My Internal Sales Distribution List</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>My Internal Support Distribution List</td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>

Add / maintain the values through the web interface. These values are stored in the InfoBurst repository.

Values

- pgrill@infoburst.com
- sbhatia@infoburst.com
- jmeloche@infoburst.com
- rpedigo@infoburst.com

Email Recipients from **External Distribution List**

Select Admin / Distribution Management / External Distribution Lists

Create / Edit a List

<table>
<thead>
<tr>
<th>List Name</th>
<th>Type</th>
<th>Details</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>MySalesManagersByEmailStateAndCityfromDB</td>
<td>Database Query</td>
<td>select distinct email from [Email Lookup] where state = '@states' and city = '@city'</td>
<td></td>
</tr>
<tr>
<td>MySalesManagersByEmailStatefromDB</td>
<td>Database Query</td>
<td>select distinct email from [Email Lookup] where state = '@states'</td>
<td></td>
</tr>
<tr>
<td>MySalesManagersDBList</td>
<td>Database Query</td>
<td>select distinct email from [Email Lookup]</td>
<td></td>
</tr>
<tr>
<td>MySalesManagersFileList</td>
<td>File</td>
<td>File C:\Info_Lists\MySalesManagers.txt</td>
<td></td>
</tr>
</tbody>
</table>
**External Distribution List (FILE)**

The file that you select must be on a Network Share that is available to InfoBurst.

```
MySalesManagersFileList
```

This shows a local file reference to the InfoBurst Server. To use a file on another server, use the UNC format for the pathname E.g. `\server\Info_Share\Lists\MySalesManagers.txt`

**External Distribution List (DB Query)**

Create the DB Connection to this database first – menu item Admin / DB connections

```
MySalesManagersDBList
```

```
Database Connections:
DB_MySalesManagers

SQL Query (use @name to create parameters):
select distinct email from [Email_Lookup]

Result Column Name:
email
```
External Distribution List (LDAP)

Firstly – run the utility program Register Instance available from the start program menu under InfoSol/InfoBurst. Configure your connection to your email server.

Now create a new External Distribution List and select the LDAP tab

Email address contained in BO report (implicit distribution list)

You need to adjust your BO report as follows: -

(A) Add a variable of name #IB#Email
(B) Create a Global Filter linking #IB#Email to it’s underlying data field
When selecting email recipients – choose the implicit EDL listed in the tree structure

**Type:** Email

- External Distribution Lists
  - MySalesManagersbyStateandCity
  - MySalesManagersbyStatefromDB
  - MySalesManagersDBList
  - MySalesManagersFileList
  - MySupportStaffFileList
- Internal Distribution Lists

**Email address contained in BO report from Excel data provider (implicit distribution list)**

Use the technique described in the previous example. This should work even if your email addresses are stored in an Excel file.
Report Tab Selection
Where possible, InfoBurst will allow you to select individual report tabs for distribution if your report has multiple tabs.

![Image of report tab selection]

Using this feature you could decide to send one report tab to recipient A in say Excel format and a second report tab to recipient B in say PDF format while a third recipient C may receive all report tabs in say HTML format.

Report Format Selection
InfoBurst allows you to choose between a series of formats as shown in the screenshot below:

![Image of report format selection]

Email in-line HTML or file attachment
InfoBurst allows you to choose between publishing HTML format reports as separate file attachments or directly into the body of the email rendering graphics and including navigational hyperlinks for navigating between report tabs.

Email Compression / Security options
InfoBurst allows you to choose ZIP compression for your file attachments. You can further choose to generate a single ZIP file attachment containing all reports or to generate a separate file attachment for each report. In addition you can choose to add security to the ZIP file by way of a ‘Password’ (the User will need to enter this in order to open the file) and also 128-bit AES data encryption to ensure that if the file is intercepted it cannot be read.

Controlling the presentation of the email
You use the email general option tab to define additional information to go along with the report in your email.
Using the Email (General) tab you can specify the Subject/From Address/Message for the emails in the burst. The radio button allows you to display the email message at the beginning or the end of the email. The checkboxes enable you to offer an Un-subscription message with the email and to choose whether to combine emails for a recipient into a single email.

The look and feel of the email is controlled by a series of HTML templates that we deliver in the following location: -

C:\Program Files\InfoSol\InfoBurst\html_templates

E.g. email_html_inline_template.txt

You can modify any of the templates to produce your own custom version. Full details of how to do this are contained in the InfoBurst Installation & Configuration Guide.
InfoBurst Macro Usage

This chapter explains the various special macros that can be used to control dynamic values in your distributions such as date, time and report parameter values.
E.g. generate a dynamic network path for report distribution purposes
There are several macro types available – these are listed below. There are a number of places where they can be used within the InfoBurst Web screens. A cross-reference chart is included at the end of this section that indicates which macro types are applicable for each usage.

Macro Types

There are eight types of macro available, distinguishable by the macro formats below. Macro references are normally specified using [] (square brackets). Note: If you need to include a square bracket as part of your naming convention then ask your InfoBurst Administrator to change the macro delimiter to a different type of bracket character.
The exceptions to this are the Date/Time Prefix/Suffix Template macro and the Command Line macro. The last two are not strictly macros, but are included here as they enable the InfoBurst User to substitute dynamic values at run-time into their bursts.

<table>
<thead>
<tr>
<th>Parameter Macro</th>
<th>Format: [=parameter]</th>
<th>Retrieve the current value of a report parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date Macro</td>
<td>Format: [MMDDYY]</td>
<td>Retrieve the system date or a calculated date (and many other variations)</td>
</tr>
<tr>
<td></td>
<td>Or extended format</td>
<td>[date expression&gt;,&lt;format string&gt;]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Retrieve a date and specify a separate format string</td>
</tr>
<tr>
<td>Fiscal Calendar Macro</td>
<td>Format: [FC:macro]</td>
<td>Retrieve a date (or a string) from a Fiscal Calendar (where macro corresponds to a column heading)</td>
</tr>
<tr>
<td>Lookup Table Macro</td>
<td>Format: [LC:macro]</td>
<td>Retrieve a string from a Lookup Table/Calendar (where macro corresponds to the Key of a Key/Value pair)</td>
</tr>
<tr>
<td>Or [LT:macro]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>System Macro</td>
<td>Format: [$SysMacro]</td>
<td>Retrieve the value of a System Variable (where SysMacro is ‘BURSTNAME’ or various others)</td>
</tr>
<tr>
<td>Function Macro</td>
<td>Format: [@function()]</td>
<td>Generate a string value from an InfoBurst variable (various options may be supplied)</td>
</tr>
<tr>
<td>Date/Time Prefix/Suffix Template Macro</td>
<td>Format: _MM_DD_YYYY</td>
<td>Retrieve a date/time for use as a filename prefix/suffix (and other specific variations)</td>
</tr>
<tr>
<td>Command Line Macro</td>
<td>Format: %B or %D</td>
<td>Pass parameters to the command line special case for passing burst name or document name to a command line program or .bat file</td>
</tr>
</tbody>
</table>
Also, there is a USER FUNCTION for combining 1 or more standard macros as a new name.

**Macro Usage**

Macros may be used for the following:

- To supply a value for a document parameter/prompt when setting up a burst
- To supply a value that is used to generate a Network Path or file name for document distribution purposes
- To supply a value that is used to generate an FTP Remote Path for document distribution purposes
- To supply a value that is used to generate a WebDAV Path for document distribution purposes
- To supply a value that is used to generate a Trigger File Name or to control the text content of the trigger file
- Supplying a value that is used to generate an attachment name override
- Supplying a value that is used to populate the text message of an email distribution
- Supplying a value that is used to populate the parameter of an EDL
- Supplying a value that is used to determine the document name when distributing a document to the BusinessObjects repository
- Supplying a value that is used to determine the Group or Username when distributing a document to the BusinessObjects repository
- Supplying a value that is used to determine the Category name when distributing a document to the BusinessObjects repository
- Supplying a value that is used to determine the document name when distributing a document to the XI Enterprise Server
- Supplying a value that is used to determine the Folder name when distributing a document to the XI Enterprise Server
- Supplying a value as a parameter to a command line executable program or batch file
Supplying a value for a Document Parameter when you set up the burst

This relates to the DOCUMENT PARAMETERS screen. The document parameter can be either a document refresh prompt or an InfoBurst filter parameter (set up in BusinessObjects as #IB#<param>)

Applicable: **Date/Time related macros (all formats supported)**
Examples:
- [TODAY] format M/D/YY
- [WW] week number
- MONTH[MM+1] next month prefixed with literal “MONTH”

Applicable: **Fiscal Calendar macros**
Examples:
- [FC:QTR_END_DATE] selected column from a Fiscal Calendar
- [FC:CALENDAR_MONTH_NAME]

Applicable: **Lookup Table macros**
Examples:
- [LC:DIVISION_CODE] selected column from a Lookup Table
- [LC:COMPANY_NAME]
Supplying a value that is used to generate a Network Pathname / FTP Remote File Path / WebDAV Network Path during burst distribution

This relates to the DOCUMENT DISTRIBUTION screen.

Applicable:  Parameter Name macro
Examples:    \svr2\reports\[=STATE]
            \svr2\reports\[=STATE]\[=CITY]
            where STATE and CITY are refresh parameters or InfoBurst filter parameters defined for the current document.

Applicable:  Date/Time related macro (all formats supported)
Example:     \svr2\reports\week_[WW].pdf

Applicable:  Fiscal Calendar macro
Example:     [FC:QTR_END_DATE] selected column from a Fiscal Calendar

Applicable:  Lookup Table macro
Example:     [LC:DIVISION_CODE] selected column from a Lookup Table

Applicable:  System Macro - Burst Name
Example:     \svr2\reports\[SBURSTNAME]

Applicable:  Function Macro – Customize the file name
Example:     \svr2\reports\[@first(REGION,2)]
            Where REGION is a refresh parameter or InfoBurst filter parameter defined for the current document.

Applicable:  Date/Time related template macro as a file prefix or suffix
Example:     _MM_DD_YYYY
            where separation characters are passed ASIS

Note:
The above macros can be used in combination when generating any Pathname – see example

Example:  \svr2\reports\[SBURSTNAME]\[=STATE]\[TODAY,MMDDYY].PDF
Supplying a value that is used to determine the name or the content of a trigger file for interfacing with your batch environment

This relates to the DOCUMENT DISTRIBUTION screen.

Applicable: **Parameter Name macro**
Examples: TRIGGER[[]STATE].GO
where STATE is a refresh parameter or InfoBurst filter defined for the current document.

Applicable: **Date/Time related macro (all formats supported)**
Example: TRIGGER[[]WW].GO

Applicable: **Fiscal Calendar macro**
Example: TRIGGER[[]FC:QTR_END_DATE]
selected column from a Fiscal Calendar

Applicable: **Lookup Table macro**
Example: TRIGGER[[]LC:DIVISION_CODE]
selected column from a Lookup Table

Applicable: **System Macro - Burst Name**
Example: TRIGGER[[]SBURSTNAME]

Applicable: **Function Macro – Customize the file name**
Example: TRIGGER[[@first(REGION,2)]]
Where REGION is a refresh parameter or InfoBurst filter parameter defined for the current document.
Supplying a value that is used to generate an attachment name override

This relates to the DOCUMENT DISTRIBUTION screen.

Applicable:  **Parameter Name macro**  
Examples:  RPT_ATTACH_[=STATE]  
where STATE is a refresh parameter or InfoBurst filter defined for the current document.

Applicable:  **Date/Time related macro (all formats supported)**  
Example:  RPT_ATTACH _[WW]

Applicable:  **Fiscal Calendar macro**  
Example:  RPT_ATTACH _[FC:QTR_END_DATE]  
selected column from a Fiscal Calendar

Applicable:  **Lookup Table macro**  
Example:  RPT_ATTACH _[LC:DIVISION_CODE]  
selected column from a Lookup Table

Applicable:  **System Macro - Burst Name**  
Example:  RPT_ATTACH _[$BURSTNAME]

Applicable:  **Function Macro – Customize the file name**  
Example:  RPT_ATTACH _[@first(REGION,2)]  
Where REGION is a refresh parameter or InfoBurst filter parameter defined for the current document.

Supplying a value that is used to generate the text of an email message

This relates to the DOCUMENT DISTRIBUTION screen.

Applicable:  **Parameter Name macro**  
Examples:  Sales Report for State [=STATE]  
where STATE is a refresh parameter or InfoBurst filter defined for the current document.

Supplying a value that is used to define an email EDL parameter

This relates to the DOCUMENT DISTRIBUTION screen for email

Applicable:  **Parameter Name macro**  
Examples:  @region=[=STATE]  
where STATE is a refresh parameter or InfoBurst filter defined for the current document.
Supplying a value that is used to determine a Document Name, a Group/Username or a Category Name when distributing a document to the BusinessObjects repository

This relates to the DOCUMENT DISTRIBUTION screen.

Applicable:  To Document Name/Group/Category

Examples:

- **Document Name**
  - Applicable:  *Parameter Name macro*
  - Examples:  
    - Document Name: Sales Global [=STATE]
      where STATE is a refresh parameter or InfoBurst filter defined for the current document.
    - Applicable:  *Date/Time related template macro*  as a file prefix or suffix
      - Example:  Document Name: Sales Global _MM_DD_YYYY

- **Document Group or Category**
  - Applicable:  *Parameter Name macro*
  - Examples:  
    - Document Group: [=STATE]
      where STATE is a refresh parameter or InfoBurst filter defined for the current document.
    - Applicable:  *Function Macro*
      - Example:  Document Category: [@first(REGION,2)]
      where REGION is a refresh parameter or InfoBurst filter defined for the current document.
Supplying a value that is used to determine a Document Name or a Folder Name when distributing a document to an XI Enterprise Server

This relates to the DOCUMENT DISTRIBUTION screen.

Applicable: **To Document Name/Folder**

Examples:

```
<table>
<thead>
<tr>
<th>Type: Business Objects XI Platform</th>
</tr>
</thead>
<tbody>
<tr>
<td>Folder:</td>
</tr>
<tr>
<td>Document Name: Sales Global</td>
</tr>
<tr>
<td>Overwrite if exists?: □ Yes</td>
</tr>
<tr>
<td>Add Date/Time stamp?: □ No, □ Suffix, □ Prefix Format: MMYY</td>
</tr>
</tbody>
</table>
```

**Document Name**

Applicable: **Parameter Name macro**

Examples: Document Name: Sales Global [=STATE]

where STATE is a refresh parameter or InfoBurst filter defined for the current document.

Applicable: **Function Macro**

Example: Document Category: [@first(REGION,2)]

where REGION is a refresh parameter or InfoBurst filter defined for the current document.

Applicable: **Date/Time related template macro** as a file prefix or suffix

Example: Document Name: Sales Global _MM_DD_YYYY
Supplying a value that is used to determine a Document Name or a Folder Name when distributing a document to an XI Enterprise Server

**Document Folder**

Applicable: *Parameter Name macro*
Examples: Document Group: [=STATE]
where STATE is a refresh parameter or InfoBurst filter defined for the current document.

Applicable: *Function Macro*
Example: Document Category: [@first(REGION,2)]
where REGION is a refresh parameter or InfoBurst filter defined for the current document.

Select folder icon generates pop-up dialog window
**Supplying a value that is used as a parameter to a command line program**

This relates to the DOCUMENT DISTRIBUTION screen.

Applicable:  

*Command Line - \%B or \%D parameters*

Examples:  

DB_RPTOK.BAT \%D

Pass the name of the current InfoBurst document to a .BAT file that is used to update an application database with a status indicator

DIST_CSV.EXE \%B

Run a program that reads data from the burst name directory and formats the report files in a User custom format
**Date/Time Macros**

These are split into DATE macros and NUMERIC macros.

The difference between them is that you can reliably specify date calculations using Date macros, whereas with Numeric macros you are able to specify date-part calculations that do not apply to the date as a whole.

For example, if today’s date is March 1st 2005

Using \[\text{LASTDAYINMONTH-1M}\] will generate February 28th 2005
Using MM-1 will generate 02 and DD-1 will generate 00

This is the full list of available macros:

<table>
<thead>
<tr>
<th>Macro</th>
<th>Description</th>
<th>Examples</th>
<th>Supports +</th>
<th>Date Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>\text{TODAY}</td>
<td>Today's Date</td>
<td>5/5/04</td>
<td>Yes</td>
<td>D/M/Y</td>
</tr>
<tr>
<td>\text{LASTDAYINMONTH}</td>
<td>Last day in current month</td>
<td>1/31/04</td>
<td>Yes</td>
<td>D/M</td>
</tr>
<tr>
<td>\text{FIRSTDAYINMONTH}</td>
<td>First day in month</td>
<td>1/1/04</td>
<td>Yes</td>
<td>D/M</td>
</tr>
</tbody>
</table>

The default format for the above DATE macros is **DD/MM/YY**.

To generate leading zeros, add "00" after the macro name (example, \text{TODAY00})

To generate leading zeros and use delimiters, add "00" after the macro name (example, \text{TODAY00})

To generate a 4-digit year, add "YY" after the macro name (example, \text{TODAY0Y})

To generate the year first, add "Y" before the macro name (example, \text{YTO0DAY})
Extended Date Macro Format

The extended format will be applicable to Date Macros (see table on previous page). You will be able to combine a date macro with a date format string to determine the date generated.

Examples: (Assuming today’s date is 03/24/05)

\[DATE-1M,YYYYMM]\nThis macro will generate 200502

\[LASTDAYINMONTH,MM_DD_YY]\nThis macro will generate 03_31_05

\[FIRSTDAYINMONTH,DAY DD/MM/YYYY]\nThis macro will generate TUESDAY 01/03/2005

Note that any characters not recognized as date template characters are passed over to the generated string as is. E.g. _ (underscore)
**Date / Time Template Macro**

Used to apply a custom date/time to the end of a filename defined for a report distribution. These dates may be defined with separation characters e.g. MM_YYYY

<table>
<thead>
<tr>
<th>Macro</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>MM</td>
<td>Month of year</td>
<td>02 or 10</td>
</tr>
<tr>
<td>DD</td>
<td>Day of month</td>
<td>05 or 12</td>
</tr>
<tr>
<td>YY</td>
<td>Current Year</td>
<td>04</td>
</tr>
<tr>
<td>YYYY</td>
<td>Current Year</td>
<td>2004</td>
</tr>
<tr>
<td>HH</td>
<td>Hour</td>
<td>12</td>
</tr>
<tr>
<td>MN</td>
<td>Minute</td>
<td>34</td>
</tr>
<tr>
<td>SS</td>
<td>Second</td>
<td>20</td>
</tr>
<tr>
<td>TIME</td>
<td>Current Time</td>
<td>12_34_20</td>
</tr>
<tr>
<td>WEEK</td>
<td>Week#</td>
<td>05</td>
</tr>
<tr>
<td>WW</td>
<td>Week#</td>
<td>05</td>
</tr>
<tr>
<td>OTR</td>
<td>Quarter</td>
<td>01</td>
</tr>
<tr>
<td>DY</td>
<td>Short Day</td>
<td>Mon</td>
</tr>
<tr>
<td>DAY</td>
<td>Long Day</td>
<td>Monday</td>
</tr>
</tbody>
</table>
**System Macros**

These are used to retrieve various system names from the InfoBurst repository.

<table>
<thead>
<tr>
<th>System Macro Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$BURST or $BURSTNAME</td>
<td>Name of the burst</td>
</tr>
<tr>
<td>$SCHEDULE or $SCHEDULENAME</td>
<td>Name of the schedule</td>
</tr>
<tr>
<td>$DOCNAME</td>
<td>Name of the document</td>
</tr>
<tr>
<td>$DOCINAME</td>
<td>Name of the document instance</td>
</tr>
<tr>
<td>$USER</td>
<td>Name of the requesting User</td>
</tr>
<tr>
<td>$OWNER</td>
<td>Name of the burst owner</td>
</tr>
</tbody>
</table>
**Function Macros**

A set of function macros is provided that will offer additional options during the construction of a pathname or other string for distribution purposes.

Syntax: `[@function()]`

Where `@function` can be any of the following:

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>@first</td>
<td>Extract first n characters from a variable</td>
<td><code>[@first(CITY,2)]</code> - retrieve the first 2 characters of the CITY parameter</td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>[@first(AGENT)]</code> - retrieve the first character of the AGENT parameter</td>
</tr>
<tr>
<td>@replace</td>
<td>Replace characters in a variable</td>
<td><code>[@replace(ADDRESS, \,-)]</code> - replace any backslashes in the address with dashes</td>
</tr>
<tr>
<td>@sp</td>
<td>Convert characters in a variable to space</td>
<td><code>[@sp(FULLNAME, _)]</code> - convert a name like JIM_ROBERTS to JIM ROBERTS</td>
</tr>
<tr>
<td>@rm</td>
<td>Remove characters in a variable</td>
<td><code>[@rm(AMOUNT, $)]</code> - remove the dollar sign from a currency value</td>
</tr>
</tbody>
</table>
**User Functions**

InfoBurst includes ‘User Functions’ that give you the ability to combine several standard function macros together into one new macro – to save repetition and time.

An example of this would be combining the values from 2 prompt parameter values to form part of a generated pathname

Reference as [@State and City]

This would return for example \Arizona\Tucson for use within the pathname.

Another example of this would be to use an ‘alias’ User Function to make it easier to reference something that already exists

Reference as [@Yesterday]
Innovative solutions for Crystal Xcelsius

InfoBurst is well positioned to add significant value to a Crystal Xcelsius deployment and can offer two innovative solutions:

1. Dashboard distribution
2. Dynamic data management

Dashboard distribution

Xcelsius dashboards (.SWF) can be cataloged with InfoBurst version 2.5 build 107 and later and added to a burst for scheduling and distribution.

When adding the file, you would typically want to check the box for “Link to this file” so that InfoBurst can always retrieve the most up to date .SWF from the network.

When creating a burst, you will have the option to use native format or HTML format, and these can be combined with most destinations. If you choose native format, just the .SWF file will be distributed on its own however it is more likely you will choose HTML format and in this case InfoBurst will create a wrapper for the .SWF object with the correct dimensions and a browser-safe mechanism to load the .SWF without requiring ‘object activation’.

If you wish to email the dashboard, choose Email/HTML and this will embed a copy of the dashboard direct into the email body with a link back to the InfoBurst server where the most recent version of the dashboard has been stored. As a dashboard is ‘active content’, you will probably have to change your email client security settings in order to see the embedded dashboard.

For XI users it is also possible to publish a hyperlink into the XI folder system so that the dashboard can be viewed with InfoView.

If the dashboard is ‘dynamic’ and uses the XML data feature of Xcelsius to connect to a web server and obtain live data, InfoBurst can do much more….it can create a ‘Dashboard Package’ that supports both on-line and off-line viewing with local history that is accumulated each time a burst is executed.

A dynamic dashboard requires a live connection to a server and there is typically no easy way to view data from another day, and if you do not have access to the server (you are out of the office) the dashboard will typically stall and return no data.

When InfoBurst creates a Dashboard Package, it will connect to the server and retrieve the XML data (takes a snapshot of the data) and create a special HTML wrapper that can use this local data as well as connect to the live server, this gives the recipient the choice to see live data or local...
historic data. If InfoBurst creates the package on a regular basis such as once a day or once a week, each deployment of the package will create a new local history file.

If you wish to distribute a dynamic dashboard outside of the company, you can also tell InfoBurst not to include a live link, in this case only the local data will be used.

**Cataloging a Crystal Xcelsius Dashboard**

Just like any other document that you wish to use with InfoBurst, the first step is to catalog the document so that InfoBurst knows its type and location.

InfoBurst can work with a Crystal Xcelsius dashboard that has been exported in the following formats:

- SWF
- PDF
- PPT

If you choose to use the PDF or PPT option, the document will be treated just like any other static file (such as Excel or Word) and the distribution will deliver the file to the chosen destination.

If you choose to use the SWF format, InfoBurst will recognize the file as a Crystal Excelsius Dashboard and will be able to offer more features such as HTML based distribution and for dynamic XML dashboards InfoBurst can create a ‘Dashboard Package’.

Using the ‘Add New Document’ menu option, browse to the location of the .SWF file and check ‘Link to this File’

Press ‘Continue’ and enter a name/description and choose the folder to store the document:
Press ‘Save’ and InfoBurst will examine the .SWF file to see if was created with Crystal Xcelsius (the .SWF file format is used by other tools such as Flash and Swish).

The document catalog screen will initially show the file as a generic SWF, but when InfoBurst examines the file it will change its type to become a Crystal Xcelsius document.

You will see this as the catalog operation starts:

After catalog has completed, you will notice the icon has changed:
Document Information

When viewing document information, you will see a new tab called ‘SWF Info’ that shows the dimensions of the dashboard and an optional preview checkbox that allows you to see the dashboard.

Width: 650
Height: 418
Dynamic XML based Dashboard Information

If the dashboard uses one or more XML data buttons, you will see the URL for each button. InfoBurst will be able to create a ‘Dashboard Package’ that allows you to view the dashboard using both local data and live data via the URL.

For more information about how InfoBurst can be used to provide Xml data for an Xcelsius dashboard, see the section “Dynamic Data Management”.

© InfoSol Inc Page : 113
Burst distribution options in detail

InfoBurst supports the following combinations for dashboard delivery:

- Native/Network
- Native/Email
- Native/FTP
- Native/InfoBurst Server
- Native/WebDav
- Native/XI Platform
- HTML/Network
- HTML/Email
- HTML/FTP
- HTML/InfoBurst Server
- HTML/WebDav
- HTML/XI Platform
- IDP/Network
- IDP/Email
- IDP/FTP
- IDP/InfoBurst Server
- IDP/WebDav

Native format options

Native format will deliver the dashboard in its original format; this could be SWF, PDF or PPT.

For Network, FTP and WebDav destinations, the file will be delivered to the desired location.

For Email, the file will be attached.

For InfoBurst Server, the file will be stored on the central storage area and a link to the file will be created. This option supports historical instances.

For XI Platform, a hyperlink will be created in the given folder that points to the .swf file on the InfoBurst server.
**Example 1 – Native distribution via Email**

Here is an example of creating a burst that delivers a dashboard via email.

Create a new burst, select the dashboard (or set of dashboards) and choose the folder to store the burst:

Next, add a new distribution and check the format is ‘Native’:
Select ‘Email’ from the Destination tab options:

This example targets a single user but you could choose one or more users/groups, enter ad-hoc email addresses or even choose a distribution list based on a file or database query.

Save the settings and complete the burst. Review the burst summary:
Execute the burst and review the activity log:

<table>
<thead>
<tr>
<th>Status</th>
<th>Time</th>
<th>End Time</th>
<th>Action</th>
<th>Detail</th>
<th>Additional Info</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>12:47:50 PM</td>
<td>12:47:38 PM</td>
<td>INIT</td>
<td>Versa2.5, Instance=INFOLIGHT, Working, Files=Infoburst_deployed Crystal Xcelsius Static ExampleSample.swf</td>
<td></td>
</tr>
<tr>
<td></td>
<td>12:47:36 PM</td>
<td>12:47:36 PM</td>
<td>RECAST_LINKED</td>
<td>Infoburst_deployedCrystal Xcelsius Static ExampleSample.swf</td>
<td>Size=213,780</td>
</tr>
<tr>
<td></td>
<td>12:47:37 PM</td>
<td>12:47:37 PM</td>
<td>OPEN</td>
<td>MyDashboard, Time=STATIC900F</td>
<td></td>
</tr>
<tr>
<td></td>
<td>12:47:37 PM</td>
<td>12:47:37 PM</td>
<td>CREATE_NATIVE</td>
<td>MyDashboard.swf</td>
<td>Size=213,780</td>
</tr>
<tr>
<td></td>
<td>12:47:37 PM</td>
<td>12:47:37 PM</td>
<td>QUEUE_EMAIL</td>
<td><a href="mailto:jwlbux@infoburst.com">jwlbux@infoburst.com</a></td>
<td>MyDashboard.swf</td>
</tr>
<tr>
<td></td>
<td>12:47:37 PM</td>
<td>12:47:37 PM</td>
<td>CLOSE</td>
<td>MyDashboard</td>
<td></td>
</tr>
<tr>
<td></td>
<td>12:47:37 PM</td>
<td>12:47:37 PM</td>
<td>SEND_EMAIL</td>
<td><a href="mailto:jwlbux@infoburst.com">jwlbux@infoburst.com</a>, Server=27.0.0.1</td>
<td>Attach 01 InlineText000 Notify 50, Total Attached Size=213,780</td>
</tr>
<tr>
<td></td>
<td>12:47:38 PM</td>
<td>12:47:38 PM</td>
<td>EMAIL_ATTACHMENT</td>
<td><a href="mailto:jwlbux@infoburst.com">jwlbux@infoburst.com</a>, File=MyDashboard.swf</td>
<td>Size=213,780</td>
</tr>
<tr>
<td></td>
<td>12:47:39 PM</td>
<td>12:47:39 PM</td>
<td>FREE_RESOURCES</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>12:47:41 PM</td>
<td>12:47:41 PM</td>
<td>END</td>
<td>End of Burst</td>
<td># Distributions Today = 0015</td>
</tr>
</tbody>
</table>

The latest version of the dashboard was taken from the linked network location and emailed as an attachment.

Note: The HTML/Email combination will in-line the dashboard so there is no attachment. See the section that describes the HTML options.
Example 2 – Native distribution to Network & InfoBurst Server

Here is an example of creating a burst that delivers the dashboard to both a network share and also stores a copy centrally on the InfoBurst server and sends an email notification via email with a link that can be clicked to view the dashboard.

Create a new distribution and select ‘Network Share’:

Enter or select a path. In this example, a share is used with a dynamic macro [$burst] to create a new folder with the name of the burst.

We now have one distribution created:

Next, create a second distribution and choose Native/Server:

Select the ‘Email Notify’ tab and select the recipients. You can choose from one or more defined users or groups or even use a distribution list based on a file or SQL query:
Save this burst and review the burst summary.

Run the burst using the ‘Execute Now’ option.

The activity log for the burst shows the steps performed, the latest version was copied from the network (Linked option) and then stored on a network share where a new folder was created (based on the name of the burst). A copy was also stored on the InfoBurst server and an email sent with a link to this file.
The file was stored on the network:

![Dashboard1](image_url)
An email was sent with a description of the document and a link:

When the link is clicked, a browser window will appear and ask for the logon details to the InfoBurst server, and then the document will be displayed:
HTML format options

The HTML format options add value over the Native format in that InfoBurst will generate a ‘wrapper’ for the dashboard that includes sizing the dashboard correctly and allowing a ‘safe’ load of the dashboard in a browser by avoiding the ‘click to activate’ mechanism. The dashboard will be ready for interaction immediately.

The HTML format is suited to Email and Server distribution, however you can deploy HTML format to the Network, FTP or WebDav locations if you wish. In this case there will be three files delivered:

- Dashboard.swf
- Dashboard.htm
- CreateSWF.js

The .htm file is the InfoBurst generated wrapper page and the .js file is a javascript file that allows for the ‘safe’ load without requiring object activation.

For XI Platform, a hyperlink will be created in the given folder that points to the dashboard.htm file on the InfoBurst server.

Example 1 – HTML distribution

Simply select HTML format instead of Native format and choose both Email and Server as before.

© InfoSol Inc Page : 122
The burst summary shows the two distributions:

When the burst executes, it will send an in-line email rather than an attachment and the server document will have a ‘wrapper’ to ensure the size is correct and also display the document name.

<table>
<thead>
<tr>
<th>Status</th>
<th>Start Time</th>
<th>End Time</th>
<th>Action</th>
<th>Detail</th>
<th>Additional Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔️</td>
<td>10:24:30 AM</td>
<td>10:24:30 AM</td>
<td>BIT</td>
<td>Ver=v2.5, instance=MPV, OPT=1, Working</td>
<td></td>
</tr>
<tr>
<td>✔️</td>
<td>10:24:30 AM</td>
<td>10:24:30 AM</td>
<td>RECATALOG_LINKED</td>
<td>C:\InfoBurst\Deploy\Crystal\Kebab\Static\Example\StaticExample.xswl</td>
<td>Size=212,780</td>
</tr>
<tr>
<td>✔️</td>
<td>10:24:30 AM</td>
<td>10:24:30 AM</td>
<td>RPT</td>
<td>MyDashboard, Type=STATIC/SWFT</td>
<td></td>
</tr>
<tr>
<td>✔️</td>
<td>10:24:30 AM</td>
<td>10:24:30 AM</td>
<td>CREATE_HTML</td>
<td>MyDashboard.html</td>
<td></td>
</tr>
<tr>
<td>✔️</td>
<td>10:24:30 AM</td>
<td>10:24:30 AM</td>
<td>HTTP_EMAIL</td>
<td><a href="mailto:nelson@infoburst.com">nelson@infoburst.com</a></td>
<td>MyDashboard [LINE]</td>
</tr>
<tr>
<td>✔️</td>
<td>10:24:30 AM</td>
<td>10:24:30 AM</td>
<td>STORE_SERVER</td>
<td>c:\infoburst\www\oroinfoburst2\Local\site\77\Create\547.js</td>
<td>Size=1,017</td>
</tr>
<tr>
<td>✔️</td>
<td>10:24:30 AM</td>
<td>10:24:30 AM</td>
<td>STORE_SERVER</td>
<td>c:\infoburst\www\oroinfoburst2\Local\site\73\MyDashboard\template</td>
<td>Size=444</td>
</tr>
<tr>
<td>✔️</td>
<td>10:24:30 AM</td>
<td>10:24:30 AM</td>
<td>STORE_SERVER</td>
<td>c:\infoburst\www\oroinfoburst2\Local\site\73\MyDashboard\template</td>
<td>Size=212,780</td>
</tr>
<tr>
<td>✔️</td>
<td>10:24:30 AM</td>
<td>10:24:30 AM</td>
<td>CLOSE</td>
<td>MyDashboard</td>
<td></td>
</tr>
<tr>
<td>✔️</td>
<td>10:24:30 AM</td>
<td>10:24:30 AM</td>
<td>SEND_EMAIL</td>
<td><a href="mailto:nelson@infoburst.com">nelson@infoburst.com</a>, Server=Infoburst 427.0.0.1</td>
<td>MyDashboard</td>
</tr>
<tr>
<td>✔️</td>
<td>10:24:30 AM</td>
<td>10:24:30 AM</td>
<td>FREE_RESOURCES</td>
<td></td>
<td></td>
</tr>
<tr>
<td>✔️</td>
<td>10:24:30 AM</td>
<td>10:24:30 AM</td>
<td>END</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Here is the email received, rather than an attachment or link, the dashboard is delivered ‘in-line’ (note – you will probably need to enable ‘active content’)

The server version looks the same:
Example 2 – Creating a hyperlink on the XI Platform

Here is an example of creating a hyperlink on the XI platform to allow the dashboard to be viewed using InfoView:

We have asked for the hyperlink to be called “Latest Sales Global Dashboard” and stored in the public folder called “QA”.

© InfoSol Inc Page : 125
**IDP format options**

The IDP format (InfoBurst Dashboard Package) is only available to dynamic XML based dashboards that are compatible with InfoBurst (see next section for more details).

The idea behind a ‘Dashboard Package’ is to be able to deliver a dynamic dashboard to people who may not always have a live connection to the server, and to provide a mechanism whereby daily/weekly historical data can be accumulated and viewed.

The IDP format is best suited to Email or Server distribution, but you can deliver to the Network, FTP or WebDav and in this case a set of files will be delivered in much the same way as for HTML, with the main difference being the generated ‘wrapper’ will allow for the selection of local history files or live data.

For Email, a self-running package is created (with an .IDP extension) that will install the dashboard and merge any existing local historical files into the display interface.

**Example – IDP distribution**

If the document has a defined XML URL, you will see the ‘Dashboard Package’ option in the list of available formats:

If you choose ‘Allow live access’, the generated wrapper will include a link to the defined URL to allow for real-time access, this would typically be enabled for internal delivery but disabled if the dashboard were to be delivered outside of the company where the URL is not accessible.
Here is the burst summary:

When the burst runs, InfoBurst will connect to the XML URL and retrieve the XML data and include it in the package.
The email will have an attached file and a short note to explain how to run the package:

Save the attached file to a temporary directory:

Next, re-name the file from IDP to EXE and check the file properties as it is likely that the email client has marked the file as un-safe:
Notice the security note at the bottom, if you do not ‘Unblock’ this file, you will be unable to run it.

After unblocking, you can simply run the file:
Next, choose a temporary location:

An installer window will appear and copy the required files to the `\Program Files\InfoSol\InfoBurst\IDP` folder and create a shortcut on the desktop to the dashboard.

When you click the shortcut, you will see a different wrapper has been created that allows you to choose between local or live data.

Clicking on the ‘Local Data’ button will load the dashboard and use the data collected by InfoBurst. Each time the package is installed, a new local history file will be deployed and the choice of local data will grow.
To see up to the minute data, click ‘Live Data’ and the dashboard will be re-loaded and will use the XML URL.
Dynamic Example (XC dashboard with XML data button)

Local Data 06/13/2006 Live Data

2004 2005 2006

Sales 2004

Q1 Sales Q2 Sales Q3 Sales Q4 Sales

Show Data

Refresh Data

idle
How to create an XML driven Dashboard compatible with InfoBurst

Using Xcelsius 4.5 Workgroup designer, create a dashboard that uses a standard data range in your Excel workbook and use the XML data button ‘Enable Load’ feature to define a named range that will replace this data.

For example, we may wish to have a selector and a graph work together to display sales data by year for each quarter.

![Dashboard Example](image)

The dashboard uses a simple data range:

<table>
<thead>
<tr>
<th>Year</th>
<th>Q1 Sales</th>
<th>Q2 Sales</th>
<th>Q3 Sales</th>
<th>Q4 Sales</th>
<th>Total Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>2004</td>
<td>6000</td>
<td>15000</td>
<td>9000</td>
<td>17000</td>
<td>31000</td>
</tr>
<tr>
<td>2005</td>
<td>6000</td>
<td>6500</td>
<td>7700</td>
<td>15400</td>
<td>35600</td>
</tr>
<tr>
<td>2006</td>
<td>6500</td>
<td>8000</td>
<td>9000</td>
<td>18500</td>
<td>42000</td>
</tr>
</tbody>
</table>

Rows 3-5 contain data for each quarter for years 2004, 2005, and 2006. A formula is used to total each row into the G column.
The Xcelsius dashboard uses a selector to choose the year and copies the data block to a working row in the spreadsheet to use as a data source for a graph and gauge.

An XML data button is used to dynamically replace the data block between C3 and F5 and uses a range called “SalesData” and uses a worksheet cell as the URL to use:

Cell B10 will contain the URL used to obtain your data values, it would normally be something like http://server/page.aspx

If you wish to see what the XML data should look like, click ‘Preview XML’
In the behavior tab, be sure to select Refresh On Load:

Next, you need to set ‘Use Flash Variables’ in the Export Settings:

Click ‘Define Variables’

Add a new variable called RefreshURL and set the Variable Selection to the same cell that is used by the XML data button URL. In this case it is going to update cell B10 (see previous page where we used this cell for the XML button).

In summary, to use an XML driven dashboard with InfoBurst, simply ensure the URL for the button uses a cell in the spreadsheet rather than a hard-coded URL and create a Flash变量 called RefreshURL that updates the same cell.
How to use the genxml.aspx web script to generate XML data

If you wish to experiment with live XML data and need some dummy data, InfoBurst ships with an XML random generator especially for Xcelsius dashboards.

The URL is http://server/infoburst/services/genxml.aspx

server is the name of your server that runs InfoBurst.

The script can be configured using parameters.

var Name of variable
rows Number of rows to generate
cols Number of columns per row
rndmax Max number for random generator
rndscale Multiplier for random value

The defaults are

var=MyVar
rows=1
cols=1
rndmax=20
rndscale=1000

Examples – assuming server is called web01:

Generate data for variable “SalesData” with 3 rows, each having 4 columns of data with each value in the range 1000 to 20000

http://web01/infoburst/services/genxml.aspx?var=SalesData&rows=3&cols=4

Generate data for variable “MyData” with 10 rows, each having 1 column of data with each value in the range 1 to 50

http://web01/infoburst/services/genxml.aspx?var=MyData&rows=10& rndmax=50& rndscale=1
To test the page first, simply enter the URL direct into your browser and you will get back the raw XML data.

http://web01/infoburst/services/genxml.aspx?var=Sales&rows=2&cols=5&rndmax=5&rndscale=5

```xml
  - <data>
    - <variable name="Sales">
      - <row>
        <column>25</column>
        <column>15</column>
        <column>15</column>
        <column>20</column>
        <column>25</column>
      </row>
      - <row>
        <column>10</column>
        <column>5</column>
        <column>10</column>
        <column>15</column>
        <column>5</column>
      </row>
    </variable>
  </data>
```

See the section “Dynamic data management” on how you can use the InfoBurst Xml Data Source feature to provide data to an Xcelsius dashboard without having to write any custom scripts.
Example Dashboards

InfoBurst ships with two example dashboards that you can use to experiment with and test the various forms of distribution.

These will be found on the InfoBurst server in the main application folder.

The examples can be loaded into the Xcelsius designer tool and include the source file for the dashboard and the Excel workbook.

Crystal Xcelsius Dynamic Example

The DynamicExample.swf is an example of a dashboard that uses an XML data button to retrieve data from a server.

The XML URL can be pointed at the genxml.aspx page described in the previous section to generate a data range for the variable “SalesData” with 3 rows and 4 columns.

Crystal Xcelsius Static Example

The StaticExample.swf is based on the above dynamic version but does not contain an XML data button.
Dynamic data management

The workgroup edition of Crystal Xcelsius offers a number of ways for a dashboard to connect and retrieve data dynamically rather than having to change and re-import the underlying Excel model and re-export the dashboard manually.

The external options include:

- XML Data Button
- Web Service
- Live Office

To use the XML option requires that you develop a custom solution to obtain and format the data. This would normally involve writing a web script to connect to a database and run some queries and then re-format the returning rows to fit in with the XML format expected by Xcelsius.

The Web Service option can connect to any existing web service and can be a good choice when the amount of data is quite small and easily structured.

The Live Office option can connect to data in a Crystal or WebI report however it requires quite a few layers of software and performance can become an issue as more and more users of the dashboard use it.

The XML option offers the best in terms of flexibility and performance but can be difficult to setup and maintain.

After talking at length with customers who have been trying to deliver Xcelsius dashboards with dynamic data, a new feature was added to InfoBurst called Xml Data Sources (XDS) that has been designed specifically to bridge the gap between your data and the Xcelsius dashboard while also providing a responsive and scalable solution.

Some of the features of XDS are:

- Can use data from a BusinessObjects or WebI report
- Can issue direct database queries
- Maps to each “range” of an XML data button
- Can create crosstabs and add row/column totals
- Can be scheduled to refresh the data
- XML data is pre-built and cached for optimal performance
XDS Workflow

To create a solution using XDS, first create an Xcelsius dashboard using a static data model so that your data visualization is working as you expect.

When creating the data model, give a thought to the blocks/ranges that will be supplied using an Xml Data button.

The basic workflow for using XDS is:

- Create an Xml Data Button in your dashboard
- Create one or more named ranges to replace data regions in your data model
- Using InfoBurst, create a new XDS
- Select the data source for the XDS (BI Report or Database Query)
- Create corresponding XDS Ranges to map to the ranges in the Xml Data Button
- For each range, select the data provider and data format (such as Crosstab)
- Configure each range to suit, setting options such as sort, number format etc
- Test each range to ensure the data is as you expect
- Execute the XDS to refresh the data and build/cache the Xml for the dashboard
- View the generated Xml and copy the URL into the Xcelsius dashboard Xml URL field
- Test the dashboard using preview mode
- Export the dashboard
- Distribute the dashboard (using InfoBurst!)
Create a new XDS

Use the Admin->Xml Data Sources menu to manage one or more XDS. You can edit an existing XDS using either the standard icon or to jump straight to the ranges for the XDS use the icon.

To create a new XDS, simply press Create New.

First, enter a name for this XDS. This name will be the “key” that identifies this XDS and will be used as a parameter to the getXDS.aspx script that will be called from the Xcelsius dashboard’s Xml Data Button.

Data Source – BI Document

InfoBurst supports the following BI documents for use with XDS:

BusinessObjects FullClient 6.x
BusinessObjects Desktop Intelligence
BusinessObjects WebI XIR2

If you select a BI document, InfoBurst will check that the document has been setup for use as an XDS data source and you will see the following message:

✔ The selected document is ready to be used as an Xml Data Source.

If the document has parameters, you will see these listed and you can supply suitable values.

If the document has not yet been configured for use with an XDS, you will see:

⚠️ The selected document needs to be re-cataloged with the Enable for use with Xml Data Source option set.

See the section “Setting up a document for use with XDS”.
Data Source – Data Query

The Data Query option will only be available if you have created and tested at least one query. To create a query, use the **Admin->DB Connections** menu to create a connection to a database and then use the SQL option to create and test a query.

Select one or more queries to use with this connection

- Conn_SQL_jwilcoxIt_1_ljwork_sa
- GetManagers
- FirSalesTest
- IB on my machine
  - Burst Info
  - GetGroups
  - NameAndEmail
- Query1

Allow real-time execution (GetXDS.aspx?k={key})

You can choose one or more queries and it is possible to use queries from different database connections so that you can bring information together from different sources.

For a Data Query there is also the option to allow **real-time execution**. This means that the getXDS.aspx URL to call InfoBurst can be called with k=$Key and InfoBurst will execute/refresh the XDS in real-time (and also store a cached copy of the XML).

If you enable this option, you need to make sure that you will not cause a long delay if the query takes some time to execute. This option should only be used when the overhead of a real-time execution is really worth it.
**Publish XML**

If you wish, you can enter a network folder and InfoBurst will create a copy of the resulting XML (called XDSName.xml) for use with off-line dashboards. This is where you could modify the XML Data Button within the Xcelsius dashboard and enter a file-name into the URL field rather than an http:// address.

**Last Refresh**

If you edit the XDS, you will see the date/time of the last refresh and you can also view the latest cached XML.

When you click on the View link, a new browser window will open and the URL will be the exact URL you will need to enter into the URL field of the XML Data Button within the Xcelsius dashboard.
Note the URL:

http://server-name/InfoBurst/services/getXDS.aspx?k=XDSName
Setting up a document for use with XDS

If you wish to use a BI report, the option to use this report as an Xml Data Source needs to be set so that InfoBurst can discover and collect information about each data provider.

To set this option, find the document in Catalog Management and select the **Data Providers** tab.

Save the document and re-catalog with **InfoBurst** so that the data elements for each data provider can be detected.

After the document has been re-cataloged, you should see a drop-down list of data elements for each Data Provider:
Create a new Range

Each XDS can have one or more Ranges and these will map to the ranges in the Xml Data button within the Xcelsius dashboard.

The Ranges screen will list any existing ranges for the XDS along with a brief description of the format, data provider and data elements:

<table>
<thead>
<tr>
<th>Select</th>
<th>Range(s)</th>
<th>Format</th>
<th>Data Provider</th>
<th>Data Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SalesbyEmp</td>
<td>CrossTab</td>
<td>Query 1 with FinSalVM</td>
<td>Employee, Customer State, Revenue</td>
</tr>
<tr>
<td></td>
<td>SalesbyEmpTots</td>
<td>CrossTab</td>
<td>Query 1 with FinSalVM</td>
<td>Customer, Customer State, Revenue</td>
</tr>
<tr>
<td></td>
<td>SalesbyState</td>
<td>CrossTab</td>
<td>Query 1 with FinSalVM</td>
<td>Order Year, Customer State, Revenue</td>
</tr>
</tbody>
</table>

Use the Create New button to create a new range.

Enter the Name to match a range in the Xml Data Button.

<table>
<thead>
<tr>
<th>XDS</th>
<th>SalesGlobal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>SalesbyEmpTots</td>
</tr>
</tbody>
</table>

On the left side of the screen you will see the available data providers and data elements that can be used.

Data Provider: Query 1 with FinSalVM

Data Elements:
- Customer
- Customer City
- Customer State
- Employee
- Number of Orders
- Order Date
- Order Month
- Order Year
- Revenue
The color of each element icon indicates the data type:

- String
- Number
- Date

On the right side you will see a series of tabs that offer the following formats:

- CrossTab
- Data Selector
- Raw Data
**CrossTab**

This is the most common type of format and will arrange and summarize 3 data elements in a familiar crosstab style.

Drag and drop data elements from the left to the blue boxes on the right that represent the X, Y and Summary values.

Here is an example:

This will generate a crosstab with **Customer State** along the X-axis and **Employee** down the Y-axis with the **Revenue** summarized for each Employee/State.

It is possible to filter by a specific data element, to do this drag and drop an element into the blue box and the available values will be displayed. Enter one or more values (separated with “;”).

**If you filter with more than one value, this will create multiple ranges and will append the filter value to the Range Name.**

For example, to filter by Order Year for 2000:

Filter:  *Note: Multi-Values will append filter value to RangeName*

<table>
<thead>
<tr>
<th>Data Element</th>
<th>Value or Value1;Value2;..</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Order Year</strong></td>
<td>2000</td>
</tr>
</tbody>
</table>

2000

If the Range is called “SalesTotals”, this will generate “SalesTotals” for the year 2000.
To filter by Order Year for the years 1999, 2000, 2001 and 2002:

Filter: Note: Multi-Values will append filter value to RangeName

<table>
<thead>
<tr>
<th>Data Element</th>
<th>Value or Value1;Value2;...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Order Year</td>
<td>1999;2000;2001;2002</td>
</tr>
<tr>
<td>2002</td>
<td></td>
</tr>
</tbody>
</table>

If the Range is called “SalesTotals”, this will generate the following ranges:

- SalesTotals1999 (for Year 1999)
- SalesTotals2000 (for Year 2000)
- SalesTotals2001 (for Year 2001)
- SalesTotals2002 (for Year 2002)
There are some options that can also be used to control how the CrossTab is generated.

**Column Totals**

Choose if you want column totals to be generated at the top or bottom (or none).

**Row Totals**

Choose if you want row totals to be generated at the left or right side (or none).

**Column Header Date Format**

Choose either a short or long date format.

**Row Header Sort Order**

By default data elements will be sorted A-Z but you can supply a list of values instead if you require a custom sort order.

**Abbreviate State names**

Check this option to convert states such as Arizona to AZ.

**Summary Values Suppress Zero**

Check this option to replace any zero values with a space.

**Summary Values Convert to %**

Check this option to convert the summary values to a percentage of the total.

**Summary Values Number Format**

Choose if the numeric values are to be converted or rounded.
Data Selector

This format will return up to 2 columns of data and is useful for supplying sorted data and summary values (such as YTD totals).

Drag and drop the required data elements into the 1st Column and 2nd Column blue boxes.

The 1st Column data element must be a string.

The 2nd Column data element must be a number.

There are some options that can also be used to control how the Data Selector is generated.

Column 1 Sort Order

By default data elements will be sorted A-Z but you can supply a list of values instead if you require a custom sort order.

Column 1 Include

By default column1 is included but you can un-check this option if you are planning to return a set of data ranges to Xcelsius and wish to “join” them together without needing the headers.

Abbreviate State names

Check this option to convert states such as Arizona to AZ.

Column 2 Suppress Zero

Check this option to replace any zero values with a space.

Column 2 Number Format

Choose if the numeric values are to be converted or rounded.
Raw Data

This format will return selected data elements “as-is”.

Select the elements of the Data Provider to include

- Data Elements
  - Customer
  - Customer City
  - Customer State
  - Employee
  - Number of Orders
  - Order Date
  - Order Month
  - Order Year
  - Revenue

You can choose one or more data elements and use the All/None buttons to quickly toggle the selected items.
Preview the range

A very useful feature is the ability to see what the generated range will actually look like and this gives you the chance to refine the options as required.

On the Ranges screen for an XDS you can click on the icon to cause InfoBurst to generate the range in real-time and display the resulting data in a scrollable data grid with a summary of the selected options:

```
Preview Range 'SalesbyState'  

Format is CrossTab
Data Elements: (')=Order Year, (X)=Customer State, (S)=Revenue
Column Totals on BOTTOM
Summary Values Numeric Format is Integer
Column Header Date Format (X) is Short Date (e.g. 10/17/2006)
Summary Values Suppress Zero
State Names will be abbreviated
```

<table>
<thead>
<tr>
<th>AZ</th>
<th>CA</th>
<th>CT</th>
<th>FL</th>
<th>GA</th>
<th>IA</th>
<th>IL</th>
<th>KS</th>
<th>MD</th>
<th>MI</th>
</tr>
</thead>
<tbody>
<tr>
<td>1999</td>
<td>272527</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2000</td>
<td>246844</td>
<td>331914</td>
<td>76581</td>
<td>230302</td>
<td>52782</td>
<td>170869</td>
<td>156995</td>
<td>100639</td>
<td>1637</td>
</tr>
<tr>
<td>2001</td>
<td>136783</td>
<td>446713</td>
<td>73710</td>
<td>123315</td>
<td>59218</td>
<td>79323</td>
<td>238616</td>
<td>74617</td>
<td>374</td>
</tr>
<tr>
<td>2002</td>
<td>320294</td>
<td>468915</td>
<td>34962</td>
<td>141294</td>
<td>83152</td>
<td>14559</td>
<td>159811</td>
<td>112300</td>
<td>15400</td>
</tr>
<tr>
<td>Total</td>
<td>705921</td>
<td>1520069</td>
<td>185253</td>
<td>494910</td>
<td>195152</td>
<td>264752</td>
<td>555820</td>
<td>287556</td>
<td>17611</td>
</tr>
</tbody>
</table>

The data used for the preview is the last set of data collected when the source document was cataloged or when the XDS was last refreshed.
Execute the XDS

The XDS needs to be “executed” like a Burst in order for InfoBurst to get the latest data from the source document (or database) and build/cache the XML that will be used by the Xcelsius dashboard.

You can execute an XDS manually by editing the XDS and clicking on the button or you can create a re-occurring schedule just like you would for a regular burst.

To create a schedule, use the Schedules->Create New Xml Data Source Schedule menu and select one or more XDS to add to the schedule.

For more details on how to create a schedule, refer to the section “Scheduling in Detail”.
Appendix A – Migrating to XIR2

For the majority of existing InfoBurst customers, the migration to BO XIR2 will involve migrating existing FullClient reports to the new Desktop Intelligence format on the XIR2 server platform.

Of course, these documents have been used in many bursts, and in turn there are many distributions configured for these documents.

InfoBurst 2.5 provides a document migration tool that will help to re-map documents within bursts to point to the new, migrated DeskI documents.

When you have migrated the documents to the BO XIR2 platform, you should catalog them using the Documents->Manage Enterprise Documents. Now, select the Admin->Migrate BO Client Documents to XIR2 option.

You will first see a description of this function:

| This function will re-map an existing Business Objects full-client document to point to the same Desktop Intelligence document on the XIR2 platform. |
| The objective is to re-map the document in all bursts where the document has been referenced. Some existing distributions may not be possible to migrate as the function/format may not be available on the new platform. |

This process has 4 steps:

1. Identify original client document(s) to migrate and select the XIR2 platform where the new documents exist

2. Locate the new documents on the XIR2 platform, checking their names and parameters (if appropriate)

3. Find all occurrences where the original document(s) are used in a burst and check each distribution to see if it can be migrated without any changes.

4. Perform the migration, changing in each burst the reference to the document to point to the new document on the XIR2 platform.
Step 1

Choose the document(s) to migrate and select the target EDP

![Diagram showing document tree and target platform]

Step 2

Identify the corresponding document, InfoBurst will search all cataloged documents on the target platform and look for a match with the same name.

**Identify document on target platform**

- ![My Documents](Migtest1) (OriginalName='Migtest1.rep')
- ![Boxi2\Desktop Intelligence](Migtest1)

**Note:** If duplicates are found on the target, you will be given a choice.

InfoBurst can only migrate a document if the name and the parameter(s) are the same.
Assuming you have a match, press Continue.

Step 3

InfoBurst will now check all of the existing distributions in each burst where this document has been used.

Results from the analysis of existing distributions

- My Documents\Migtest1
  - My Burst\ExampleBurst1
    - Migtest1
      - (PDF/Network)
        - InfoBurst conversion is not supported. Will convert to provider conversion
    - Migtest2
      - (Native/BusObj Repository) Will convert to XI Platform (disabled). Need to review

Document Distribution Summary: OK to migrate = 1, Warnings = 2, Errors = 0

- My Burst\ExampleBurst1, (Excel/FTP) InfoBurst conversion is not supported. Will convert to provider conversion
- My Burst\ExampleBurst2, (Native/BusObj Repository) Will convert to XI Platform (disabled). Need to review

Actions — The following documents are ready to be migrated:

- My Documents\Migtest1 • BOX\Desktop\Intelligence\MigTest1

Most distributions can be migrated; however some will require some changes after the migration. In the above example, the document is used in 2 bursts and has 3 distributions.

The OK symbol ☑ indicates that this distribution will be migrated with no changes.

The warning symbol ⚠ indicates that migration is supported but you need to review and possibly change some details after the migration.

The abort symbol ❌ indicates that migration is not possible. You need to review the distribution and possibly delete it before migration can continue.
Migration Rules

Distributions that CANNOT be migrated

- Printer

This will need to be removed before migrating the document

Distributions that will migrate but need to be configured and enabled

- BO Repository

Will be converted to XI Folder, need to supply folder + enable

Distributions that will be converted to closest equivalent

- Excel format using InfoBurst Conversion

Will be converted to standard Excel format. Any macros or base workbooks will not be used.

For each document with no errors, you can now perform the document migration. Press Continue and the document(s) will be migrated in all distributions.
Appendix B - Excel Macro Guidelines

This chapter describes the best practices for using an external Excel macro to format a workbook and how to deal with potential run-time errors.

For EXCEL report distribution, InfoBurst supports Macro invocation when publishing to a base workbook using InfoBurst EXCEL conversion. Macros are written using VBA. Currently, if a macro is invoked, and an error occurs, this is likely to raise an interactive dialog box, and this causes the InfoBurst burst executing as a batch job to hang.

Implemented Solution
InfoBurst has been modified to check a return code from the Macro. If the return code indicates an error, then InfoBurst will trap this, and indicate an error (including a User generated error message if supplied) in the burst activity log. In order to achieve this, and stop EXCEL macros from hanging, InfoBurst Administrators need to ensure that any macros are written to conform to the following rules: -

Firstly - Here is some example code

```
Function Format_Macro() As Variant
On Error GoTo errorlabel
...
Range("A1").Select
...
...
Format_Macro = ""
Exit Function
errorlabel:
Format_Macro = "ERROR" & err.description
End Function
```

So the rules are: -
Code the macro as a **Function** and not a Sub
Specify a return parameter as a Variant
Include an On Error trap as the first statement in the code
Return a blank value if the macro executes to end
Return a string starting with “ERROR” if an error is encountered
(this is what InfoBurst will check for to indicate an error)

Notes
If you code a macro as a function inside of EXCEL, it is not listed when you select TOOLS / MACRO / MACROS from the menu bar. However, you can still invoke it as a Macro.
Existing macros coded as SUBs are still supported, but you cannot return an error to InfoBurst, and they are still liable to cause a hang unless you add in the ON ERROR trap.
Appendix C – System Diagrams

Catalog Management

InfoBurst Schema Diagram v2.5
InfoBurst Schema Diagram
v2.5
Document Distribution (Delivery Mechanism)

InfoBurst Schema Diagram

v2.5

- XI PLATFORM
  - Folder
  - Document Name
  - Overwrite Check
  - Report Name Suffix
  - E-Mail Notification

- BUSOBJ REPOSITORY
  - Domain Name
  - Document Name
  - Group
  - Category
  - Overwrite Check
  - RPT Name Suffix
  - E-Mail Notification

- INFOBURST WEB SERVER
  - E-Mail Notification
  - History

- NETWORK SHARE
  - Existing Share
  - New Share
  - Execute a Command
  - Compression (optional encryption)
  - RPT Suffix
  - E-Mail Notification

- WEBDAV
  - Location
  - RPT Suffix
  - E-Mail Notification
  - Compression (optional encryption)

- FTP SERVER
  - Secure Option
  - Compression (optional encryption)
  - RPT Suffix
  - E-Mail Notification

- NETWORK / LOCAL PRINTER
  - E-Mail Notification

- E-MAIL ATTACHMENT
  See E-Mail Notification
Document Distribution (Report Format, Email Notification, Burst Options)

InfoBurst Schema Diagram

v2.5

© InfoSol Inc
Appendix D – Glossary of Terminology

ACTIVITY LOG
A Log file in the InfoBurst Repository that keeps a record of all executed BURSTS. The Log file will confirm all bursts being executed & indicate whether they executed successfully and show a date & time stamp against all execution steps.

ADMINISTRATOR
A USER who has all access rights for administration purposes and is able to setup and manage other USERS and to access the AUDIT LOG.

AUDIT LOG
A Log file in the InfoBurst Repository that keeps a record of all actions taken by a USER or an ADMINISTRATOR through the Web Interface.

BURST
A BURST refers to a set of DOCUMENT instances that refer to physical files or documents held within a document repository. When a BURST is executed, these documents are REFRESHED with new data if required, and distributed in a number of different ways and in different formats to selected recipients. A BURST may be part of a SCHEDULE, and executed on a regular basis, or it may be executed on demand.

CATALOG
This refers to a Menu screen which lists all or selective DOCUMENTS for the current User. From this screen, DOCUMENT entries may be edited and their VISIBILITY changed.

CATEGORY
A CATEGORY refers to a logical grouping of DOCUMENTS. The purpose of a CATEGORY is to enable an ADMINISTRATOR to selectively view a list of DOCUMENTS in the CATALOG management screen, (the Filter By option) and may be used in conjunction with a FILTER. It is also available as a Filter By option when selecting DOCUMENTS that are included in a BURST – from the BURST MANAGEMENT menu option.

DISTRIBUTION TEMPLATE
A Distribution Template is something that may be created by an Administrator or a User. The purpose of a template is to provide a convenient way of setting up a frequently used distribution, and applying this at document or burst level. This helps in two ways (1) It cuts out much duplication of effort in setting up distributions, and (2) it means that potential changes to a distribution can be applied in one place, at the template level, rather than to individual distributions.

DOCUMENT
A logical document, identified by a DOCUMENT NAME and a DESCRIPTION, and including a physical location that represents the ‘deliverable’ set of data.
DOCUMENT INSTANCE
The object that represents the ‘instance’ of a document within a burst. A burst may contain more than one instance of the same Business Objects document – each instance may refer to a different Parameter set – thus resulting in several different physical documents.

EDL
An EDL or External Distribution List is a way of generating a list of dynamic email addresses for use in document distribution. An EDL may pull email addresses from many sources including from a database table via an SQL Query (Oracle/SQL Server), a flat file stored on the network file system or an LDAP (Lightweight Directory Access Protocol) Server such as Microsoft Exchange Server.

EPL
An EPL or External Parameter List is a way of generating a list of dynamic parameter values from a SQL query for use in refreshing or filtering a BusinessObjects or Crystal Reports document OR for use in controlling a conditional distribution of a document using values retrieved from a SQL query.

EVENT
An Event is a mechanism to provide conditional execution of a SCHEDULE based upon the completion status of an external application. An Event is implemented as a TRIGGER or as a FILE_EXISTS file existence check.

FILE_EXISTS
A way of implementing an EVENT based upon the existence of a file on a Network Share. This file may be deleted by the scheduler as part of an automatic reset option.

FOLDER
Used for organizing and selectively viewing objects such as documents, bursts, schedules and templates.
A folder structure is created automatically within InfoBurst version 2. Additional folders may be created, and folders moved and deleted by individual Users. Folders such as ‘Public’, may only be managed by ADMINISTRATORs.

GROUP
A logical set of USERS, identified by a GROUP NAME, established by an ADMINISTRATOR for purposes of group access to documents, bursts, and schedules and/or multi-recipient report distribution.

IDL
An IDL or Internal Distribution List is a list of email addresses that can be used with email distribution. The values are maintained within the InfoBurst repository and edited through the web user interface.
IPL
An IPL or Internal Parameter List is a list of parameter values for use in refreshing or filtering a BusinessObjects or Crystal Reports document OR for use in controlling a conditional distribution of a document. The values are maintained within the InfoBurst repository and edited through the web user interface.

LDAP
Lightweight Directory Access Protocol – as supported on various server software e.g. Microsoft Exchange Server. InfoBurst supports the use of LDAP to import users or as a source for an EDL.

MACRO
A placeholder available to the InfoBurst User when setting up document distribution. A Macro is expanded dynamically at run time to generate a Date/Time string for use as a parameter or a pathname.

REPORT BURSTING
This refers to the ability of InfoBurst to take a BusinessObjects or a Crystal document, and to refresh the document dynamically from a list of values (supplied as IPL or EPL lists), in order to create a multiple set of reports. InfoBurst includes 5 powerful features to extend report bursting to provide the ultimate in flexibility:

- The option to automatically process every section of the document generated at runtime (RPL) in place of using a driving list of values (IPL or EPL).
- The option to specify filter parameters (#IB#<parameter-name>) in place of refresh parameters, to minimize the number of times a document has to be refreshed.
- The option to provide conditional distribution (destination and format) for each section produced within a document.
- The option (for email distribution) to incorporate an External Distribution List to send sections of a report to specific email recipients.
- For email distribution, the option to obtain the target email address directly from the document itself, as long as this is set up as a global filter within the BusinessObjects document. In this case, each email address will appear in a section of its own.

RPL
An RPL or Report Parameter List is a way of ensuring that every section of information generated within a BusinessObjects document is processed for distribution. An RPL is used in place of IPL or EPL if appropriate for Customer document distribution requirements. An RPL can also be used linked to an EDL in a special case where email distribution is required and the email address is obtained directly from the BusinessObjects document.
SCHEDULE
A SCHEDULE consists of one or more BURSTS, and is executed under the control of the InfoBurst Server taking into account EVENTS and Date and Time conditional constraints. A schedule is only ever valid for today’s date. Schedules are rebuilt by the system at midnight for the following day. Any schedule which is added or amended through the web interface is re-scheduled for today according to any scheduling events and/or date & time constraints.

SCHEDULER
This corresponds to the IBSERVER Windows System Service. This is responsible for establishing whether a burst is ready for scheduling; dispatching bursts using IBRUN; cataloguing provider instances.

SERVER DOCUMENT
A document that exists within the InfoBurst repository (a Web document).

SYSTEM FUNCTION
This is a housekeeping function available to the ADMINISTRATOR for managing the log files that are accumulated within InfoBurst.

TRIGGER
One way of implementing an EVENT, this corresponds to a Boolean value stored in the InfoBurst repository. It may be set through the command line interface or through the web interface, and may also be reset by the SCHEDULER.

USER
An individual, identified by a USER NAME, who is either – a) responsible in varying degrees of authority for administering the system Or b) allowed to logon to the Website to retrieve reports Or c) a passive recipient of reports.

USER FUNCTION
A new macro that has been created to perform a specific task such as return the date for yesterday in a special format. These new ‘functions’ can be called using the [@function] syntax.

USER PROFILE
A profile applying to a User whereby Password, Email Address, Home Page & Business Objects logon details may be maintained by the User themselves.

XDS
An Xml Data Source for a Crystal Xcelsius dashboard. Data can be retrieved from either a BI report or a number of database queries and formatted into a number of ‘Ranges’ that correspond to Ranges in an Xml Data Button within the dashboard. The XDS can be scheduled and the resulting Xml data is cached for use with a web script on the InfoBurst server that is called from the Xml Data Button (using the URL).
# Appendix E - Macro Cross Reference Chart

<table>
<thead>
<tr>
<th>Macro Usage by Macro Type</th>
<th>PARAMETER Macro [=STATE]</th>
<th>DATE Macro [MMDDYY]</th>
<th>Fiscal Calendar FC:</th>
<th>Lookup Table LT:</th>
<th>SYSTEM Macro $system</th>
<th>FUNCTION Macro $function</th>
<th>DT/TM file Prefix/Suffix MM YYYY</th>
<th>COMMAND Parameter %B or %D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Document Param /Prompt Value</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Network Share Pathname</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Command Line Parameter</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Trigger File Name</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trigger File Content</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Email Text Message</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Email EDL parameter</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Attachment Name</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FTP Remote File Path</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>WebDAV Network Path</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>BO Repository Document name</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BO Repository Group/Username</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BO Repository Category name</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>XI Doc Name</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>XI Folder Name</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: You can use a **User Function** wherever a ✓ is displayed

© InfoSol Inc Page : 169