



## Getting the most out of the query builder

If you have used Visual Intercept for any period of time you know the Query Builder provides a simple way to build and save queries for practically any kind of Visual Intercept data. What is sometimes overlooked is the ability to build more sophisticated queries in the Query Builder. When combined with other features such as the \$PROMPT macro, the flexibility and power of the Query builder can really save you time and effort when querying for incident data.

Below are some of the queries we have collected that perform specific tasks beyond the normal queries a user would use. To use these queries, just download the queries file and unzip them into your Queries directory whose location is specified in the Tools|Options menu on the integration tab.

Note: All queries listed below using \$PROMPT may not be executed in Visual Intercept Web Relay or Visual Intercept Web Enterprise. The current implementation of \$PROMPT provides an interactive query mechanism suited only for Visual Intercept Enterprise desktop applications.

### [Download Now](#)

#### Available Queries:

[Incidents registered within the last N days](#)

[Incidents that are old but still active](#)

[Incidents with particular words in the subject field](#)

[Incidents by incident ID](#)

[Incidents with a particular source file attached](#)

[Incidents registered in a portion of the project hierarchy](#)

[Incidents related to other incidents in a particular project](#)

[Incidents assigned to current user with related incidents of a particular type](#)

[Incidents associated with a particular Company](#)

[Incidents associated to a contact by last name](#)

[Incidents that are still active related to an accounts contacts](#)

Incidents registered within the last N days:

File: incdstart.imf

This query fetches a list of incidents that were entered within the last N number of days specified. The \$PROMPT macro will ask how many days back you are interested in finding entered incidents for. If you wanted to view all incidents entered in the past ten days you would type in the number 10 and click OK.

Syntax:

```
Incident.IncidentID IN (SELECT Incident.IncidentID FROM Incident WHERE DATEDIFF(DAY, Incident.StartDate, GetDate()) <$PROMPT("Incidents Registered Since a Particular Day", "Please enter the number of days.", ""))
```

Incidents that are old but still active:

File: incdold.imf

This query fetches a list of incidents that have never passed through a status that would indicate that they were resolved or that work was in progress within the last N number of days specified. The \$PROMPT macro will ask how many days back you are interested in finding unworked on incidents. If you wanted to find incidents that have not been resolved for more than thirty days you would enter the number 30 and click OK.

Syntax:

```
Incident.IncidentID IN (SELECT Incident.IncidentID FROM Incident WHERE DATEDIFF(DAY, Incident.StartDate, GetDate()) >$PROMPT("Old Active Incidents", "Please enter the number of days.", "")) AND Incident.IncidentID IN (Select IncidentID FROM Incident_History WHERE Incident_History.Status NOT IN ('Closed','Development', 'Q/A','Rejected','Duplicate')) AND Incident.Status NOT IN ('Closed','Development', 'Q/A','Rejected','Duplicate')
```

Incidents with particular words in the subject field:

File: incdsub.imf

This query uses the Visual Intercept \$PROMPT macro to execute a keyword search against the Subject field. The value box will be pre-formatted with the necessary quotes and wild card characters. This query can easily be adapted for any other text field used in Visual Intercept.

Syntax:

```
Incident.IncidentID = $PROMPT("Incident by ID","Please enter an IncidentID", "")
```

An Incident by Incident ID:

File: incdid.imf

This query uses the Visual Intercept \$PROMPT macro to execute a search by IncidentID. This query is very helpful for quick searches for particular Incidents where the IncidentID is known.

Syntax:

```
Incident.IncidentID = $PROMPT("Incident by ID","Please enter an IncidentID", "")
```

Incidents with a particular source file attached:

File: incdsource.imf

This query is used to find incidents that are associated with a particular source file. This query uses the \$PROMPT macro to let you specify a particular file and searches for the file name as a keyword.

Syntax:

```
Incident.IncidentID IN (SELECT IncidentID FROM Incident_Sources WHERE Incident_Sources.Name LIKE $PROMPT("Incidents by Source File","Please enter the name of a source file","%"))
```

Incidents registered in a portion of the project hierarchy:

File: incdproj.imf

This query is used to fetch all Incidents registered against a project and all child projects specified in the \$PROMPT macro. This query is very usefull when you want to review the incidents contained in a general portion of the project hierarchy. For ease of use, the the query incorporates the \$PROMPT macro and searches for the Project name as a keyword.

Syntax:

```
Incident.ProjectName LIKE $PROMPT("Search for Incidents by Project", "Please enter a Project name.", "%")
```

Incidents related to other incidents in a particular project:

File: incdprojrelat.imf

This query is useful when you want to quickly find all Incidents related to Incidents in the Projects you are currently working in. So, for example, if you related Incidents from another project to Incidents you are currently working on because they were blocking your progress, this query would allow you to quickly review all of the related Incident that were relevant to your work. This query uses the \$PROMPT macro and searches for the Project name as a keyword.

Syntax:

```
Incident.IncidentID IN (SELECT RelatedIncidentID FROM Incident_Incidents WHERE Incident_Incidents.IncidentID IN (SELECT IncidentID FROM Incident WHERE ProjectName LIKE $PROMPT("Search for Related Incidents by Project", "Please enter a Project name.", "%"))))
```

Incidents assigned to current user with related incidents of a particular type:

File: incdmyrelat.imf

This query allows you to quickly fetch all incidents related to other incidents you are currently working on where they are related in a particular way. A particular example would be to find all incidents that are defined as "Duplicates" of the ones you are currently working on. This query uses the \$USERID macro to substitute the currently logged on user in addition to the \$PROMPT macro.

Syntax:

```
Incident.AssignID = $USERID AND Incident.IncidentID IN (SELECT IncidentID FROM Incident_Incidents WHERE RelatedType LIKE $PROMPT("Relationship Type", "Please enter the relationship type", "%"))
```

Incidents associated with a particular Company:

File: incdcomp.imf

This query fetches incident based on the company name entered in the Company field of a contact document. This query would be used to fetch all incidents related to members of a particular organization to understand the kinds of problems they have been experiencing.

Syntax:

```
Incident.IncidentID IN (SELECT IncidentID FROM Incident_Contacts WHERE Incident_Contacts.ContactID IN (SELECT ContactID FROM Contact WHERE CompanyName LIKE $PROMPT("Incidents by Company Name", "Please enter a company name.", "%")))
```

Incidents associated to a contact by last name:

File: incdcont.imf

This query would fetch the same incidents seen in the Incidents tab of a specific Contact document. However, it is very convenient way to simply see incidents related to particular person to understand the kinds of problems the contact has been experiencing.

Syntax:

```
Incident.IncidentID IN (SELECT DISTINCT IncidentID FROM Incident_Contacts WHERE Incident_Contacts.contactID=(SELECT ContactID FROM Contact WHERE Contact.Lastname LIKE $PROMPT("Search for Incidents by Contact", "Please enter a Contact name.", "%")))
```

Incidents that are still active related to an accounts contacts:

File: incdcontacc.imf

This query fetches all incidents related to contacts in a particular account. This query would be used to fetch all incidents related to an account to understand the kinds of problems they have been experiencing.

Syntax:

```
Incident.IncidentID IN (SELECT DISTINCT Incident_Contacts.IncidentID FROM Incident_Contacts WHERE Incident_Contacts.ContactID IN (SELECT DISTINCT Account_Contacts.ContactID FROM Account_Contacts WHERE Account_Contacts.AccountCode = "v1.01 beta")) AND (Incident.Status NOT IN ("Q/A", "Closed", "Duplicate", "Suspended", "Rejected", "Not Reproducible") ) AND Incident.Priority NOT IN ("low")
```