

Creating new Enterprise View tiles

Enterprise View is a customizable dashboard where users can go and set up their own customized page showing the information being collected by the 1E products. Enterprise View tiles are defined using a set of XML files located at:

```
%installdir%\1E\EnterpriseView\Dashboard.Web\Source\
```

All files should end with the extension .xml and are validated by a set of XML schemas located at:

```
%installdir%\1E\EnterpriseView\Dashboard.Web\Source\Schemas
```

On this page:

- [Specifying a data source](#)
- [XML format tile definition file](#)
- [DashboardTile tag](#)
- [Tile commands](#)

Specifying a data source

The tiles require data sources to display anything useful. The data sources must be added to the `DataSources.xml` file in the `EnterpriseView\Dashboard.Web\Source` directory. By default the `DataSources.xml` file contains four data sources: `Dashboard`, `NightWatchman`, `NightWatchmanServer` and `LOCAL`.

The `Dashboard` data source relates to the Enterprise View settings, the `NightWatchman` and `NightWatchmanServer` relate to the data held by `NightWatchman Management Center` and `NightWatchman Server Edition` products. The fourth default data source called `LOCAL` refers to a local directory where you can store and retrieve XML data.

Data source types

There are two data source types, `SQL` and `XML`. `XML` data sources refer to `XML` format files held locally in the `EnterpriseView\Dashboard.Web\Source\LocalData` directory. `SQL` data sources must have their `SQL` connection information defined.

Using your own database

If the database you are using as a data source is local to the Enterprise View website you need to add `NT AUTHORITY\NETWORK SERVICE` to the database users and grant that account the appropriate permissions for any specific tables you want to use in the database.

If the data source database is remote you need to add the machine account for the computer hosting the Enterprise View website. This will be in the format `<DOMAIN> <MACHINENAME>$` where the `<DOMAIN>` is the domain where the host computer is running and `<MACHINENAME>` is the name of the computer hosting the Enterprise View website.

For example, given a server called `ACMESVR030` running on the domain `ACME`, you need to add the following machine account as a user on the database:

```
ACME\ACMESVR030$
```

Creating an SQL data source

When creating an `SQL` data source, you need to specify the connection string to the database. This includes the server name, the database and configuration settings for the connection.

For example to create a data source for an `ACME` database residing locally to the Enterprise View website you would add the following to the `DataSources.xml` file as a new element of the `DataSources` element.

```
<DataSource Name="ACME">
  <ConnectionProperties DataProvider="SQL">
    <Value>
      <![CDATA[Server=(local);Database=ACME;Trusted_Connection=True;pooling=false]]>
    </Value>
  </ConnectionProperties>
</DataSource>
```

The XML LOCAL data source

If you want to use an `XML` file as the input data source for a tile, use the `LOCAL` data source already defined. All data files should reside in the folder, relative to the Enterprise View installation directory:

```
EnterpriseView\Dashboard.Web\Source\LocalData
```

When specifying the command in the tile using the LOCAL data source, give the name of the file (see above) containing the data.

Defining an XML data file

The XML data file needs to obey the following format. The file must start with the `<?xml version="1.0" encoding="utf-8" ?>` tag:

```
<?xml version="1.0" encoding="utf-8" ?>
```

There must be a single `<Result></Result>` tag which contains all the rows of data. Each row of data must appear in a `<Row>` tag. Each column in the row must be defined as a number of `column/value` pairs separated by spaces in the following format:

```
ColumnName1="Value1" ColumnName2="Value2"
```

For example, the following shows an XML data file representing a four column data table where the columns are named `Id`, `Month`, `PCSaving` and `ServerSaving`.

```
<?xml version="1.0" encoding="utf-8" ?>
<Result>
  <Row Id="0" Month="Sep 09" PCSaving="25399.00" ServerSaving="2500" />
  <Row Id="1" Month="Oct 09" PCSaving="23872.00" ServerSaving="2389" />
  <Row Id="2" Month="Nov 09" PCSaving="24977.00" ServerSaving="2700" />
  <Row Id="3" Month="Dec 09" PCSaving="24293.00" ServerSaving="3214" />
  <Row Id="4" Month="Jan 10" PCSaving="24556.00" ServerSaving="2847" />
  <Row Id="5" Month="Feb 10" PCSaving="25675.00" ServerSaving="2837" />
</Result>
```

XML format tile definition file

The contents of the tile definition file must start with the `<?xml version="1.0" encoding="unicode" ?>` tag:

```
<?xml version="1.0" encoding="unicode" ?>
```

DashboardTile tag

The tile definition must be placed in the `<DashboardTile></DashboardTile>` tag. The attributes for this tag lets you define general attributes for the tile, including the schemas for any presentation elements the tile definition uses and the tile's title, description and type. The following parameters can be set in the `DashboardTile` tag:

| Parameters | Notes |
|----------------------|--|
| Activator | The activator specifies how data is actively retrieved for the tile. Each activator corresponds to the tile presentation elements that will be used in the tile. There are four supported activators to choose from but only one can be set per tile. The activators and corresponding presentation elements are: <ul style="list-style-type: none">• <code>StandardListViewActivator</code> – required when standard tables are used in the tile definition.• <code>StandardSingleValueActivator</code> – required when one of the single value controls is used in the tile definition.• <code>GaugeActivator</code> – required when a gauge is used in the tile definition.• <code>StandardChartActivator</code> – used for the chart presentation elements in the tile definition. This activator is the default so charts may be used. |
| Description | Description for the tile as it appears in the configuration page of the Enterprise View website. |
| RefreshFrequencyMins | Duration (in minutes) between refreshing the data in the tile. |
| ThumbnailSource | Icon for the tile as it appears in the configuration page of the Enterprise View website. |
| Title | Title for the tile as it appears at the top of the tile and on the configuration page of the Enterprise View website. |
| Type | Must be <code>Standard</code> to enable the supported presentation elements described below. |

| | |
|-----------------------------|--|
| <p>xmlns:<object></p> | <p>The schema definitions that are required to allow particular components to be used in a tile definition. Each xmlns line added to the DashboardTile tag sets a schema definition for a particular object reference. For example, to enable the use of resources to support internationalization, add:</p> <pre data-bbox="423 233 1463 302">xmlns:res="http://www.le.com/2009/DashboardResources.xsd"</pre> <p>to the DashboardTile element. This uses the label res which must then be used when adding the resource definitions</p> <pre data-bbox="423 394 1463 464"><res:ResourceDictionary Language="en-US"></pre> |
|-----------------------------|--|

Tile commands

Each tile command is used to gather information from a particular data source. You can have a number of commands defined in a single command collection each referring to the same or a different data source. All the commands must be placed in a single <CommandCollection></CommandCollection> tag, usually as the first element inside the DashboardTile tag.

The command definition must provide the name for a data source defined in the DataSources.xml file and an identifier that will be used to run the command in a presentation element. The command identifier must be unique within the tile definition but this is not checked by Enterprise View – you will need to ensure this is the case.

Each command must be defined using the <Command></Command> tag using the following format:

```
<Command DataSource="DataSourceName" Identifier="CommandID">
  <Text>
    Command...
  </Text>
</Command>
```

where DataSourceName and CommandID is the data source and command identifier respectively and Command... is the command to run.

SQL commands

It is usual, though not required, to put the command definition in a CDATA statement. This allows XML reserved characters to be used in the command without the need for escape characters.

For example, the following command uses the ACME data source and selects the Rating and Number columns from the dbo.UserSatisfaction table in the database the ACME data source references. The command would then be used in a presentation element by referring to the ACMECMD identifier.

```
<Command DataSource="ACME" Identifier="ACMECMD">
  <Text>
    <![CDATA[ SELECT Rating, Number FROM dbo.UserSatisfaction ]]>
  </Text>
</Command>
```

XML commands

These commands always use the LOCAL XML data source and the command definition always consists of the name of the file containing the required data.

For example, the following command references the DepartmentCost.xml file which must be located in the LocalData directory. The command is then used in a presentation element by referring to the DEPTCOST identifier.

```
<Command DataSource="LOCAL" Identifier="DEPTCOST">
  <Text>DepartmentCost.xml</Text>
</Command>
```