XML Collector file conventions and formats

This section includes the following topics:

About the XML Collector file

The Collector file is an XML file with one or more rows that are generated from user-defined definitions. These data files, collected by the Add-on AppTier Collector, are put in a Collector directory in the *<i3_root>* for local client harvesting.

The Collector file location is `[i3]/products/SDK/tech_code`.

The location file is specified in the ETD file which is part of the Add-on AppTier installation package, described in Appendix A. A Collector file has the following properties:

- File name and location
- File format and contents (file header and file body)
- File closure and stamp
- File restrictions

A Collector file has the following naming convention:

tech-code_instance-name
_monitored-server_unique-id.[prf|avl]

The following table shows the elements of a Collector file name.

Table 1 Elements of a Collector file name

Element	Description
Tech-code	2-letter technology code (U1, U2, and so on). The technology code is assigned to your Add-on AppTier application during registration and is specified in the ETD file.
Monitored-server	Name of monitored (instance) server (for example: _MyServer). The monitored server is installed through AdminPoint.
Instance-name	Name of instance (added previously through AdminPoint).
Unique-id	A sequence number that is generated by the Insight SDK Collector and ensures the file name is unique. There is no standard or restriction to the sequence number range and format.
File extension	Prf is used for performance data; avl is used for availability data (not currently supported).

The body of the Collector file contains multiple row elements, each of them representing a summary of a distinct activity, collected after a default 15-minute timeslice. The entities and counters are defined as child elements with text values.

The following is one example of a GMS Collector file.

```
Filename: U1_MyServer_1104643728.xml
<rowset savvy="MS"instance="MailServer1" server="toi-vm-1">
    <row>
        <D> 2007-07-1108:15:00.0 </D>
        <msg> change request </msg>
        <sender> Eliza Doll </sender>
        <C_IP> 10.1.1.4 </C_IP>
        <type> mail </type>
    </row>
</rowset>
```

The following is the basic hierarchy for an XML-based Collector file:

- The root element <rowset> has three attributes
- Multiple <row> elements under the root; each row represents all invocations of a unique activity during a 15-minute time slice.

The following table specifies the attributes for the root <rowset> element.

Table 2 Root <rowset> element attributes

Rowset Attribute Name	Description
Savvy	A 2-letter technology code (U1, U2, and so on). The technology code is assigned to your Add-on AppTier application during registration, and is specified in the ETD file.
Instance	Instance name (added earlier through AdminPoint).
Server	Monitored (instance) server name. For example: _MyServer. The monitored server is installed through AdminPoint.

About XML Collector file rows

The body of an XML Collector file is a series of <row> elements. Each row specifies the activation of a unique activity in the Add-on AppTier during a 15-minute time slice.

The row element has sub-elements that specify values for entities and counters. The sub-element name is the same as the data sub-element's attribute collection-id value, for entities, counters, and the timestamp field in the ETD file.

The following example shows how attributes in the ETD file relate to XML tags in the SDK Collector file.

Three consecutive dots (...) represent additional standard code not given.

```
<etd version="1.0">
    ...
    <entities>
        <!-- mail msg, activity -->
        <entity eid="A" tid="501" ...>
            <data class="activity" dbcolumn="MAIL_MSG" collection-id="msg" />
            ...
    </entities>
    ...
    <misc-fields>
        <field tid="999">
            <data class="timestamp" dbcolumn="TIMESTAMP" collection-id="D" />
        </field>
    </misc-fields>
</etd>
<rowset savvy="MS" instance="MailServer1" server="toi-vm-1">
    <row rownum="1">
        <D> 2007-07-11 08:15:00.0 </D>
        <msg> change request </msg>
        ...
    </row>
</rowset>
```

All row sub-elements must be specified as a collection-id for one of the following: entity, counter, or timestamp field in the ETD file.

Observe the following standards for Collector file rows sub-element text values.

- **Timestamp field**. Values should be specified by GMT time zone. Mandatory format: YYYY-MM-DD-HH24:MM:SS
- **Entities**. Values should be enclosed with the following: <! [CDATA [….]] >
- **Counters**. Service time counters must be reported in the following format: sss.mmm where seconds and milliseconds with a decimal point have a character dot.

About file closure and stamp

The SDK Collector file cannot be harvested to the Precise FocalPoint server until the file is closed and stamped by the SDK Collector. The SDK Collector closes and stamps the file to signal file readiness for subsequent processing and loading to the PMDB.

In an ARM collection mode, closing and stamping is handled by the Insight SDK Collector. In a file collection mode, the third-party SDK Collector needs to handle these tasks.

To stamp the Collector file, the SDK Collector must create an empty file (file with size zero [0] bytes) with the same name as the XML Collector file but with a different extension.

The following are XML Collector file extensions:

- .pok extension for .prf
- .aok file for .avl

The following are file name examples that are produced by the SDK Collector for the GMS Add-on AppTier that is installed on server MailServer-1 and monitors instance MailInstance-1.

MS_MailInstance-1_MailServer-1_1189403.prf

where file size is 135 KB.

MS_MailInstance-1_MailServer-1_1189403.pok

where file size is 0 KB.

About time slices and activity summaries

Collector files are harvested at time periods having fixed intervals. These time intervals are called time slices and their length is fixed at 15 minutes. The Precise harvester waits one minute, after a time slice has passed, and searches for new files to move to the Precise FocalPoint server.

For example: at noon, harvesting starts at the following times: 12:01, 12:16, 12:31, and 12:46.

The Precise harvester searches for and processes only stamped Collector files (such as a couple of identically named .prf and .pok files) and only in the location specified in the ETD file.

The harvesting schedule just mentioned is similar to the one that takes place for an Insight Collector and for component Collectors. The Collectors usually produce one file that includes a summary of all recorded activity. "Summary" is used here to mean the aggregation of performance calculations for all invocations for the same activity during the time slice.

For example: all invocations for a Windows calculator program on the same computer, by the same login user will be aggregated into one <row> element in an OS Collector file. The timestamp in the aggregated row will point to the start of the time slice. This standard is used to optimize the Collector's footprint on the Operating System, to reduce the Collector file's size and handle the issue of scalability.

The Insight SDK Collector follows this standard when in ARM collection mode. However, this standard is not mandatory for SDK Collector file mode. The Collector can create as many files in the same time slice and use whatever aggregation mechanism needed. The SDK Collector is responsible to manage such issues as footprint, file size, and scalability.