

# Hyades Session Management

## Revision history

Rev.	Date	Author	Summary of Changes
0.1	August 23, 2004	H.M. Nguyen	First draft

## Important note

This is only one chapter of the overall “Hyades Protocol Specification” document. Reviewers are assumed to have read that document prior to this section.

## Hyades session management

This section describes the concept of session in Hyades and explains how it works and how the HCE manages and maintains session information.

This design description focuses only on how one HCE defines, creates and manages session. Similarly, the Hyades client can manage its own set of session info when dealing with multiple HCEs.

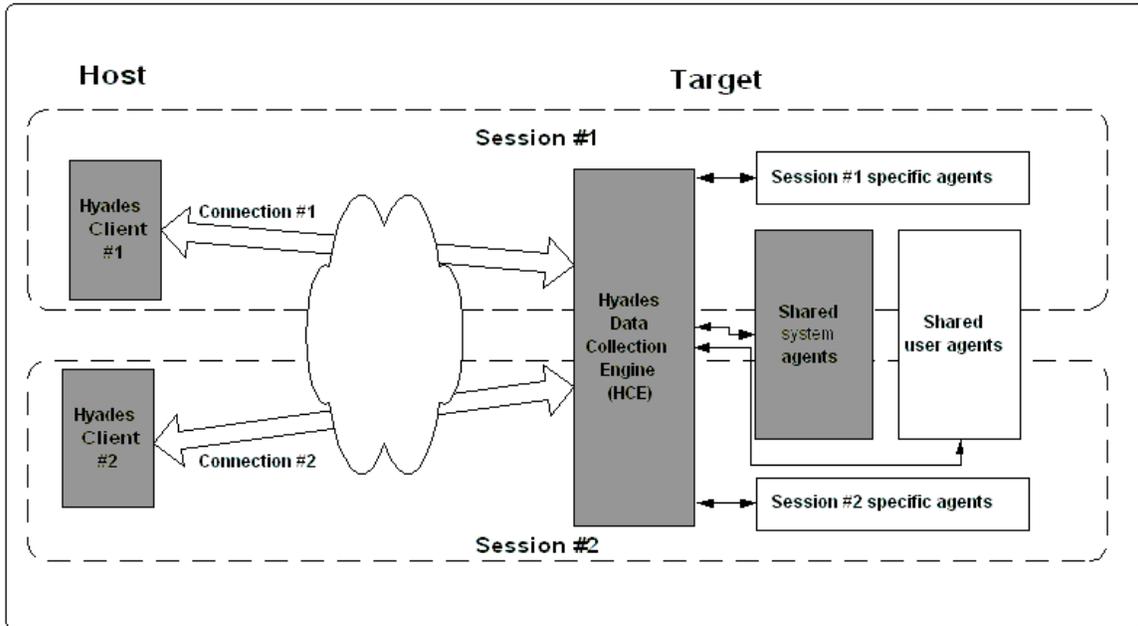
## *Terminologies and definition*

Term	Description	Usage
Connection	A link between two endpoints such as: <ul style="list-style-type: none"> <li>- Client and HCE</li> <li>- HCE and HCE</li> <li>- Agent and HCE</li> </ul>	Examples of connection include: <ul style="list-style-type: none"> <li>- Socket connection</li> <li>- Named pipe connection</li> <li>- HTTP connection</li> </ul>
Session	A logical connection	Session provides <ul style="list-style-type: none"> <li>- the mechanism to maintain state information</li> <li>- contexts to support multiple users across the same connection</li> <li>- Enablement for persistence support across multiple connections or restart</li> </ul>
Connection ID	A unique integer number assigned for a connection.	This number is unique only within the current instance of HCE.

Term	Description	Usage
Session GUID	A globally unique token string represents for this session.	This string is uniquely identifying this session globally. It can be kept persistent across restart or among different HCEs.
Session ID	A unique integer number represents for this session.	This number is unique only within the current instance of HCE. This maps 1-1 with the Session GUID and embedded in the message for performance.

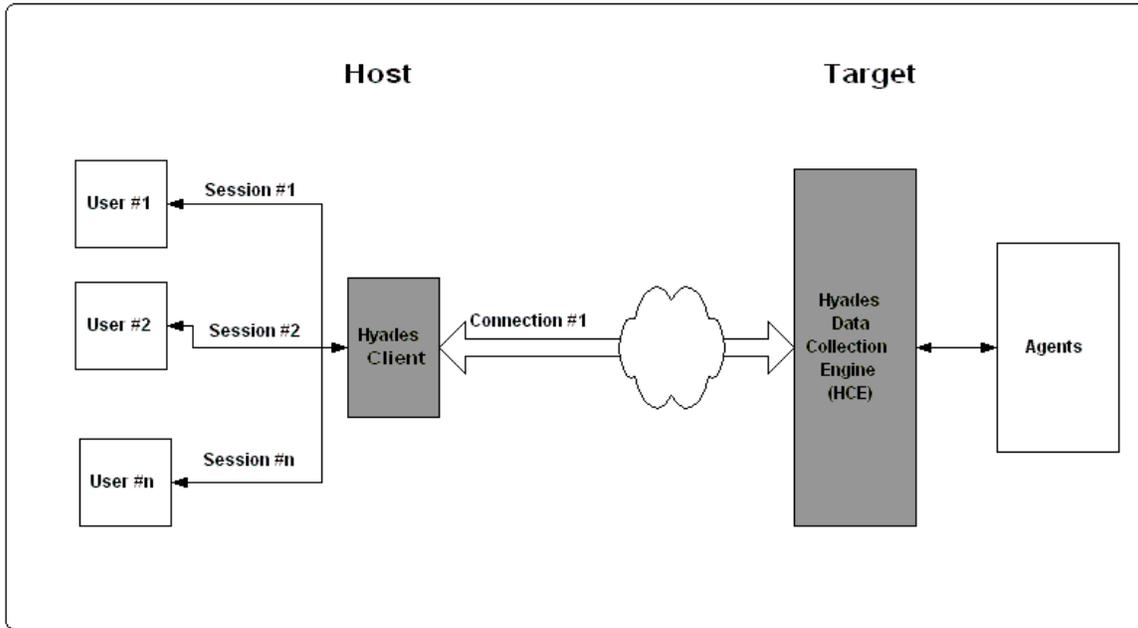
**Session concepts and usage**

**Figure 1** below shows a basic usage of a session. It typically consists of one client connection, one HCE and a set of related agents. System agents are normally shared and used by multiple sessions. It is up to the users to define and implement the user agents to be able to be shared or used exclusively by each session. Each session will require at least one connection for the data exchange.



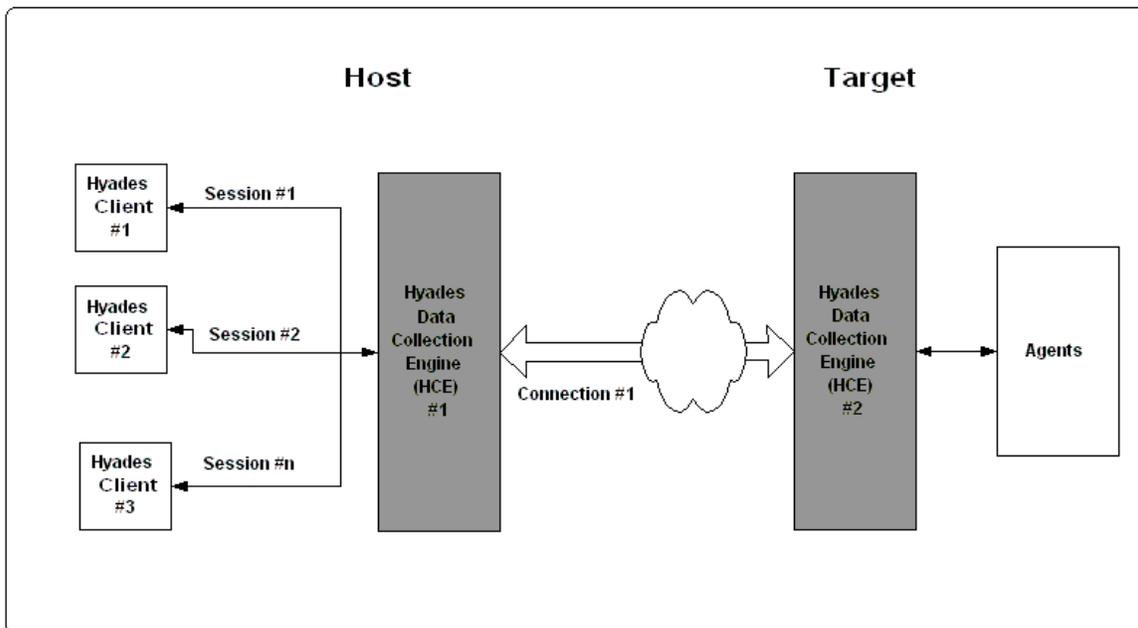
**Figure 1 – Multiple Hyades Server Sessions**

**Figure 2** below shows how the client application can make use of session information to support multiple users over the same connection.



**Figure 2 – Multiple sessions over the same client-HCE connection**

Similarly, **Figure 3** below shows how the HCE makes use of the session information to allow multiple Hyades clients communicates with remote agents of other HCEs. The HCE will need to establish only one connection to the other HCE and use it for different types of requests.



**Figure 3 – Multiple sessions over the same HCE-HCE connection**

**“Sessionable” interface and its commands**

The following commands are supported through the Hyades “Sessionable” interface. Clients and agents use these commands to establish session and manage the unit of work under that session.

Command:

`CID_CREATE_SESSION`

Data:

Bytes 0-3	Flags indicate <ul style="list-style-type: none"> <li>- whether a known session token is given (used in the case of re-establishing with an existing session)</li> <li>- whether to force the creation of the new session or not</li> </ul>
Bytes 4-7	Timeout value. 0 means “no timeout.”
Bytes 8-11	Optional string length of the session globally unique token
Remaining bytes	Session globally unique token (GUID)

Response:

`CID_CREATE_OK`

Response Data:

Bytes 0-3	Session id integer number (unique for this current HCE instance and 1-1 mapping to the session globally unique token)
Bytes 4-7	String length of session GUID
Remaining bytes	The value of the session GUID

Description:

Clients (or agents) may use this command to create a new session with the HCE over the current connection.

Optionally, the client can give an existing session GUID to re-establish the connection with an existing session.

The HCE will return the session token as well as the unique integer number representing that session. This number is mapped 1-1 with the session GUID and used to communicate with the HCE from this point on for this session.

Command:

`CID_END_SESSION`

Data:

Bytes 0-4	Session id number
-----------	-------------------

Response:

`CID_END_OK`

Response Data:

Bytes 0-4	Status code
-----------	-------------

**Description:**

Clients (or agents) may use this command to end an existing session with the HCE.

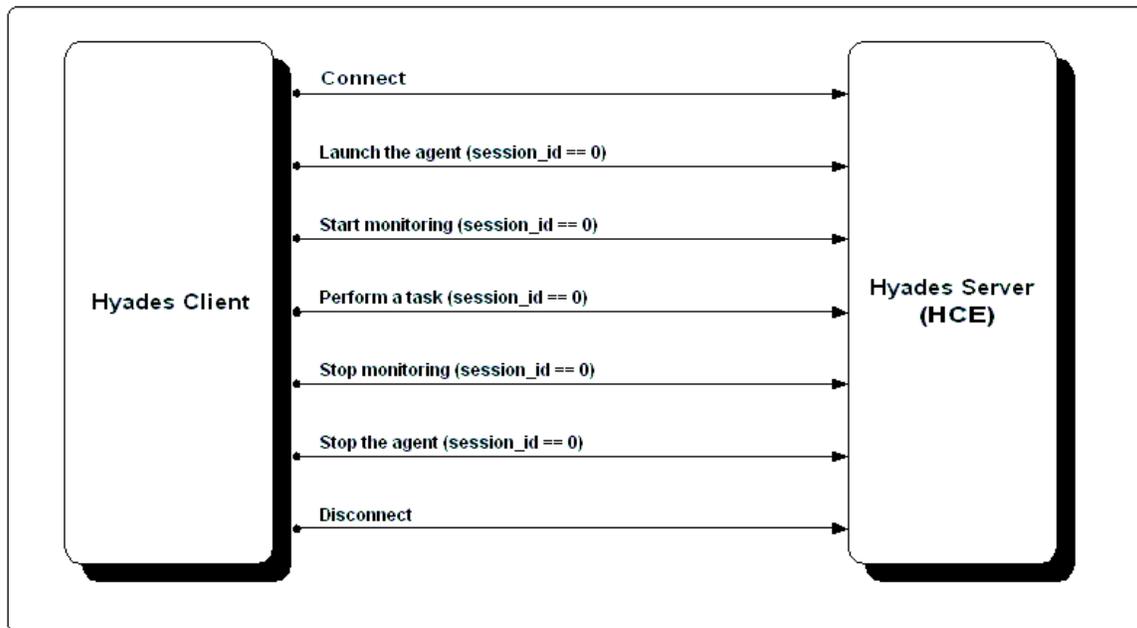
***Session tracking and how it works*****The basics of session management**

Here is the list of basic assumptions in session management:

- Security (authentication and authorization) is performed at the connection and not at session level.
- Session id is a required part of every message header. Consequently, related messages (or requests) can be correlated, audited or logged under a specific unit of work, a transaction or client activities separately for each session.
- Session id of 0 is the default. It translates into “stateless” requests to the HCE. This helps keep the client application simple where it does not have to explicitly program or involve with session management if it has no such requirement.

**An example of “stateless” default session info processing**

**Figure 4** below shows how the client can simply submit stateless requests (similar to HTTP requests) to the HCE without worrying about session information. There is no additional overhead of processing if the client does not make use of session information.



**Figure 4 – Example of default (stateless) session**

## An example of explicit session info processing

**Figure 5** below shows how the client can create and submit requests within a single session. The HCE will manage the association (mapping) between the session id and the unique GUID string that assigns to a specific session.

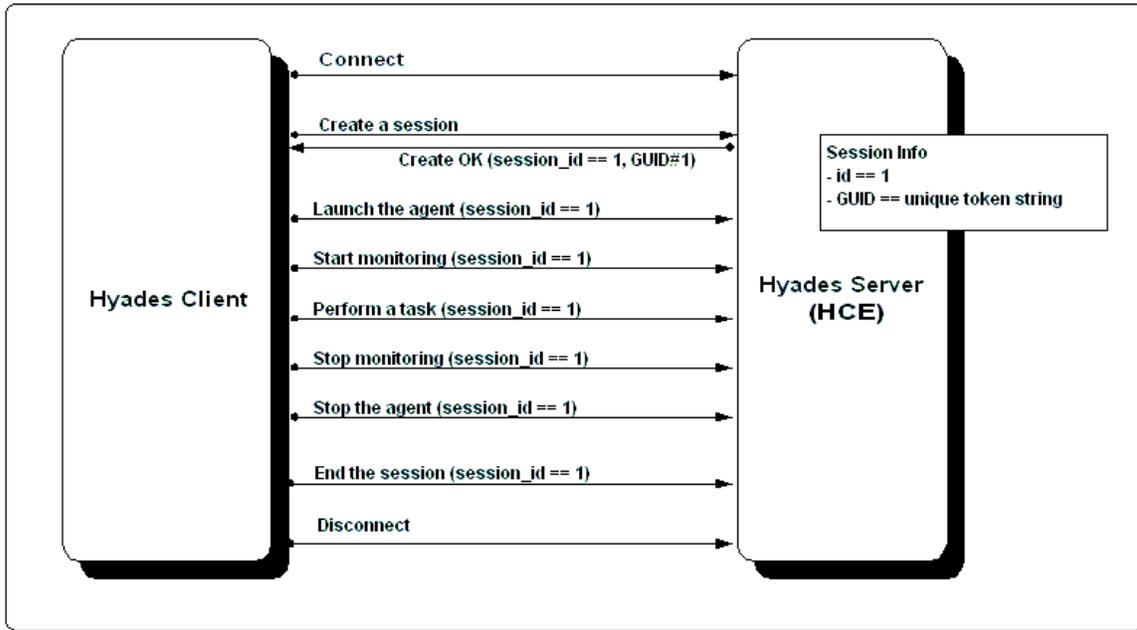


Figure 5 – Example of session creation and management

## Summary

Session information is made available for each message so that the client application has the option to make use of the context and manage its tasks at more granular level and also enable the HCE to be more scalable by sharing the same connection.

By reserving the session id of 0 as the default and making it “stateless” request, we have kept the logic of the client application in its simplest form.

However, if the session is created explicitly, there will be a globally unique token string (GUID) assigned for that session. This is basically the infrastructure that enables the HCE to support persistent session (by persisting and re-establishing the same session token across restart or multiple connections). Similarly, the HCE will also be able to make use of this session GUID to support session failover and load balancing in the future.